

DISTRIBUTION CONTENT GENERATING METHOD, METHOD AND DEVICE FOR DISTRIBUTING CONTENT, AND CODE CONVERTING METHOD

Publication number: JP2001028585

Publication date: 2001-01-30

Inventor: IDEHA YOSHIHARU

Applicant: SONY CORP

Classification:

- international: G06F12/00; G06F13/00; G06F17/30; H03M7/30; H04L29/06; H04L29/08; H04N7/16; H04N7/173; G06F12/00; G06F13/00; G06F17/30; H03M7/30; H04L29/06; H04L29/08; H04N7/16; H04N7/173; (IPC1-7): H04N7/173; H04L12/18; G06F12/00; G06F13/00

- European: H04L29/08N1; G06F17/30T; H03M7/30; H04L29/06C5; H04L29/08N5; H04N7/16

Application number: JP19990198483 19990713

Priority number(s): JP19990198483 19990713

Also published as:

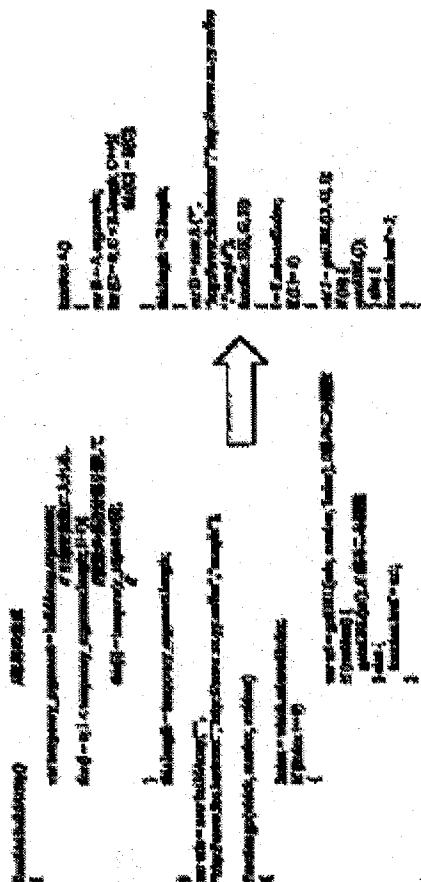
EP1115207 (A1)
WO0105038 (A1)
US7308508 (B1)
US2008028101 (A1)
EP1115207 (A4)

more >>

Report a data error here

Abstract of JP2001028585

PROBLEM TO BE SOLVED: To improve transmission efficiency of contents distribution composed of text data by replacing the character of character string o high redundancy, extracted from a script with the character or character string of low redundancy. **SOLUTION:** The script which is to be a processing object is scanned, and the character strings of high redundancy are extracted and listed. The first element (namely an original function name or parameter name) of the list is extracted, this element is made into new function name or parameter name by applying application characters and the correspondence relation of elements and application characters is preserved on a correspondence table. It is discriminated whether the application characters reach Zn, namely, reach a character string composed of Z only and when the discriminated result is positive, the number of application characters is increased by one according to the ascending order of alphabets. Processing is started to the next element of the list by incrementing (i) by 1, when all the elements on the list are completely registered on the correspondence table, the redundant character string such as function name or parameter name in the original scrip is converted to the application characters of low redundancy, and the size of the script is compressed.



Data supplied from the esp@cenet database - Worldwide

【特許請求の範囲】

【請求項1】配信コンテンツを生成する方法であって、配信コンテンツを構成するモジュールの1つとして作成された複数の文字又は文字列からなるスクリプトの内部を検索して、冗長性の高い文字又は文字列を抽出するステップと、スクリプト中で抽出された冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換えるステップと、を含むことを特徴とする配信コンテンツ生成方法。

【請求項2】置換後の配信コンテンツを配信する前に一時蓄積するステップを含むことを特徴とする請求項1に記載の配信コンテンツ生成方法。

【請求項3】前記の冗長性の高い文字又は文字列は関数名や変数名を含むことを特徴とする請求項1に記載の配信コンテンツ生成方法。

【請求項4】前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換えるステップでは、冗長性の低い文字列としてのシステム予約語の使用を排除することを特徴とする請求項1に記載の配信コンテンツ生成方法。

【請求項5】さらに、配信コンテンツの受信側で実行される処理に関与しない文字又は文字列からなる文を抽出するステップと、該抽出された文を削除するステップとを含むことを特徴とする請求項1に記載の配信コンテンツ生成方法。

【請求項6】前記の配信コンテンツの受信側で実行される処理に関与しない文字又は文字列からなる文は、所定のデリミッタで区切られたコメント文であることを特徴とする請求項5に記載の配信コンテンツ生成方法。

【請求項7】さらに、前記の冗長性の高い文字又は文字列を抽出するステップで抽出された文字又は文字列の各々についてのスクリプト中での出現頻度を積算するステップを含み、

前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列に置き換えるステップでは、出現頻度の大きい文字又は文字列ほどより文字数の短い文字又は文字列に置き換える、ことを特徴とする請求項1に記載の配信コンテンツ生成方法。

【請求項8】複数のデータ・モジュールで構成されるコンテンツを配信するコンテンツ配信方法であって、複数の文字又は文字列からなるスクリプト言語形式で記述されたデータ・モジュールの内部を検索して、冗長性の高い文字又は文字列を抽出するステップと、スクリプト中で抽出された冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換えるステップと、

置換後のデータ・モジュールを蓄積するステップと、蓄積されたデータ・モジュールを配信するステップと、を含むことを特徴とするコンテンツ配信方法。

【請求項9】前記の冗長性の高い文字又は文字列は関数

名や変数名を含むことを特徴とする請求項8に記載のコンテンツ配信方法。

【請求項10】前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換えるステップでは、冗長性の低い文字列としてのシステム予約語の使用を排除することを特徴とする請求項8に記載のコンテンツ配信方法。

【請求項11】さらに、配信コンテンツの受信側で実行される処理に関与しない文字又は文字列からなる文を抽出するステップと、該抽出された文を削除するステップとを含むことを特徴とする請求項8に記載のコンテンツ配信方法。

【請求項12】前記の配信コンテンツの受信側で実行される処理に関与しない文字又は文字列からなる文は、所定のデリミッタで区切られたコメント文であることを特徴とする請求項11に記載のコンテンツ配信方法。

【請求項13】さらに、前記の冗長性の高い文字又は文字列を抽出するステップで抽出された文字又は文字列の各々についてのスクリプト中での出現頻度を積算するステップを含み、

前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列に置き換えるステップでは、出現頻度の大きい文字又は文字列ほどより文字数の短い文字又は文字列に置き換える、ことを特徴とする請求項8に記載のコンテンツ配信方法。

【請求項14】複数のデータ・モジュールで構成されるコンテンツを配信するコンテンツ配信装置であって、複数の文字又は文字列からなるスクリプト言語形式で記述されたデータ・モジュールの内部を検索して、冗長性の高い文字又は文字列を抽出する手段と、

スクリプト中で抽出された冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換える手段と、置換後のデータ・モジュールを蓄積する手段と、蓄積されたデータ・モジュールを配信する手段と、を含むことを特徴とするコンテンツ配信装置。

【請求項15】前記の冗長性の高い文字又は文字列は関数名や変数名を含むことを特徴とする請求項14に記載のコンテンツ配信装置。

【請求項16】前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換える手段は、冗長性の低い文字列としてのシステム予約語の使用を排除することを特徴とする請求項14に記載のコンテンツ配信装置。

【請求項17】さらに、配信コンテンツの受信側で実行される処理に関与しない文字又は文字列からなる文を抽出する手段と、該抽出された文を削除する手段とを含むことを特徴とする請求項14に記載のコンテンツ配信装置。

【請求項18】前記の配信コンテンツの受信側で実行される処理に関与しない文字又は文字列からなる文は、所

定のデリミッタで区切られたコメント文であることを特徴とする請求項17に記載のコンテンツ配信装置。

【請求項19】さらに、前記の冗長性の高い文字又は文字列を抽出するステップで抽出された文字又は文字列の各々についてのスクリプト中での出現頻度を積算する手段を含み、

前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列に置き換える手段では、出現頻度の大きい文字又は文字列ほどより文字数の短い文字又は文字列に置き換える、ことを特徴とする請求項14に記載のコンテンツ配信装置。

【請求項20】コンピュータ処理に関する意味を持つ複数の文字又は文字列で構成されるソース・コードを変換する方法であって、

ソース・コード中を検索して、冗長性の高い文字又は文字列を抽出するステップと、

抽出された冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換えるステップと、を含むことを特徴とするコード変換方法。

【請求項21】前記の冗長性の高い文字又は文字列は関数名や変数名を含むことを特徴とする請求項20に記載のコード変換方法。

【請求項22】前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換えるステップでは、冗長性の低い文字列としてのシステム予約語の使用を排除することを特徴とする請求項20に記載のコード変換方法。

【請求項23】さらに、ソース・コード中で該コードの実行に関与しない文字又は文字列からなる文を抽出するステップと、該抽出された文を削除するステップとを含むことを特徴とする請求項20に記載のコード変換方法。

【請求項24】前記のコード実行に関与しない文字又は文字列からなる文は、所定のデリミッタで区切られたコメント文であることを特徴とする請求項23に記載のコード変換方法。

【請求項25】さらに、前記の冗長性の高い文字又は文字列を抽出するステップで抽出された文字又は文字列の各々についてのスクリプト中での出現頻度を積算するステップを含み、

前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列に置き換えるステップでは、出現頻度の大きい文字又は文字列ほどより文字数の短い文字又は文字列に置き換える、ことを特徴とする請求項20に記載のコード変換方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、衛星デジタル放送やマルチメディア・コンテンツの通信など、デジタル・データ配信技術に係り、特に、コンピュータ言語の命令

で構成されるデジタル・データを配信する技術に関する。

【0002】更に詳しくは、本発明は、任意に定義可能なタグを使用するマークアップ言語（例えば、XML（eXtensible Markup Language））形式のデジタル・データを配信するデジタル・データ配信技術に係り、特に、いわゆるスクリプトなどのテキスト・データからなる配信コンテンツの伝送効率と受信側での実行効率を向上するデジタル・データ配信に関する。

【0003】

【従来の技術】放送の技術分野において、データのデジタル化が急速に進められている。デジタル信号の方が、アナログ信号よりも安定性に優れ、データ圧縮率も高い。また、デジタル信号は、ケーブル、地上波、衛星波を問わず、一定の帯域でアナログ信号よりも多くのチャンネルを確保することができる。また、従来の情報伝達においては文書、音声、映像の各情報は全く別物であったが、放送データをデジタル化することにより分け隔てなく伝送することができる。

【0004】例えば、デジタル・データであれば、EPG（Electric Program Guide：電子番組ガイド）など各種のシステム情報等を映像情報や音声情報とともに送ることによって、ユーザ（視聴者）へのサービスを向上させることができる（EPGは、放送番組の放送スケジュールや番組名などの提示情報の他、VTRへの録画予約などの機能を包含する）。

【0005】デジタル形式のデータは、汎用コンピュータ・システムなどテレビジョン以外の情報機器との親和性も高い。例えば、汎用コンピュータ・システムに衛星放送用チューナ・カードを取り付けて、受信したEPGデータをコンピュータ内で解析して、番組表をコンピュータ・ディスプレイ上にウィンドウ表示して、番組切替や録画予約などをマウス・カーソルによる操作で実現することができる。勿論、放送データをコンピュータの内蔵ハード・ディスクにそのままデジタル録画することも可能である。

【0006】また、帯域を活用して、映像や音声などの放送番組本体以外のデータを送信することにより、放送番組の対話性を高めることができる。例えば、択一式のクイズを提供するような放送番組であれば、映像・音声データとともに解答を併せて送信することにより、衛星テレビ放送受信機（すなわち視聴者）側では、コンピュータ・ディスプレイ上に解答メニュー・ボタンを用意し、マウス・カーソルなどのポインティング・デバイスによる解答操作に応答して答合わせを行うことができる。

【0007】さらに対話性が高まれば、家庭内のテレビ受信機は、単に映像コンテンツを扱うだけでなく情報管制塔として飛躍することが期待できる。例えば、テレビ

受信機は、インターネット端末や電子商取引端末にもなり得る。

【0008】日本では、ARIB（電波産業会）が中心となって、デジタル衛星データ放送に関する標準化作業が進められている。これによれば、デジタル衛星データ放送では、衛星放送番組本体を構成する映像と音声のデータ（AVデータ）の他に、放送番組に付随するデジタル伝送データが並行して配信される。より具体的には、MPEG（Motion Picture Experts Group）2など所定の圧縮方式で圧縮されたAVデータと、デジタル伝送データとを多重化して構成される「トランスポート・ストリーム」（後述）の形式で、放送波として伝播される。

【0009】デジタル伝送データに載せられる放送番組情報の一例は、先述のEPG（Electric Program Guide：電子番組情報）である。また、放送番組情報は、放送番組本体のタイトルや日付、番組のキャスティングなど放送番組本体に関する固有の情報を含むことができる。また、放送番組情報が含むデータの種類やデータ構造は、番組本体が提供するサービス内容に応じてある程度系統化される。例えば、料理番組における献立や食材に関する情報や、選挙速報番組における時々刻々更新される得票状況、プロ野球中継における各選手の打撃・投球に関する個人成績やチーム順位などである。

【0010】放送番組情報は、デジタル衛星データ放送受信システム（以下、単に「受信システム」とする）側では、放送番組本体を表示出力するディスプレイ・スクリーンの一部の領域を使用して表示される。この受信システムは、通常は、放送波を受信・選局・デコードする受信機（セット・トップ・ボックス：STB）と、表示出力するテレビジョンとで構成され、一般家庭内に設置される。

【0011】ここで、デジタル伝送データとして配信されるコンテンツ（以下、「配信コンテンツ」とする）の構造について、図13を参照しながら説明しておく。

【0012】図13に示すように、配信コンテンツは、テキスト・データの他、静止画、動画、音声などの各種様々なモノメディア・データと、これら各々のモノメディア・データを統合的すなわちマルチメディアとして取り扱い、放送番組情報の有様を規定する表示・出力制御プログラム（以下では、「マルチメディア符号化アプリケーション」とも呼ぶ）とで構成される。表示・出力制御プログラムには、各モノメディア・データに対する参照（リンク）情報を組み込むことができる。

【0013】先述したARIBによるデジタル衛星データ放送の標準化作業では、この配信コンテンツとしてのマルチメディア符号化アプリケーションを記述する形式（フォーマット）として、当初はMHEG（Multi media and Hypermedia Expe

rt Group）の利用が検討されてきた。MHEGは、データ属性を定義する記述言語の一種であり、マルチメディア・コンテンツをテレビに表示し、ビデオ・オン・デマンド（VOD）やデジタル・テレビ放送において視聴者が所望の情報を引き出すような用途を想定している。日本のCSデジタル放送（SKY Perfect V）のように、既にMHEG-5が使用されている例もある。

【0014】しかしながら、MHEGは、符号化空間が固定的であり拡張性に欠ける。すなわち、一意に定められた符号化空間のみを用いて記述しなければならないので、例えば関数（API（Application Programming Interface）など）を一つ変更するのにさえ、相当の修正作業を要する。また、MHEGは、一般的な認知度が低く、MHEGで記述されたデータ・コンテンツがあまり流通しておらず、この意味において汎用コンピュータとの親和性が比較的低い。

【0015】そこで、ARIB（先述）では、MHEGに代わって、XML（eXtended Markup Language）をベースとしたデータ放送の標準化作業が検討されている。

【0016】XMLは、タグの定義が任意、すなわち属性の記述の仕方に制約がないので、自由度が高く、また、汎用コンピュータやインターネットとの親和性が高いことなどが、MHEGよりも有利な点として挙げられる。また、XMLは、次世代インターネットの記述言語としての策定も進められている。

【0017】XMLをデジタル衛星放送の標準言語で用いた場合、デジタル放送用データを、コンピュータやテレビジョン、電話機など、多種多様な情報機器間で情報交換を行うことができる。また、XMLは、タグの属性を任意に定義できることから、レイアウト指定に特化したHTMLに比しデータ処理に強い。このため、電子商取引を含め、各分野への利用も急速に進みつつある。

【0018】XML文書において任意なタグの設定が許容することは、言い換えれば、文書中に書かれた文字列を意味の付けられたデータとして扱うことを目的とするものである。すなわち、タグの定義により、タグで区切られた各タグ・データを、単なる表示目的以外の意味を持ったデータとして表現することができる。さらに、タグの構造を定義することで、XML文書又は該文書中のデータを構造化して記述することが可能となる。

【0019】既に述べたように、ARIBでは、XMLをベースにしてデジタル衛星データ放送の標準化を進めている。より具体的には、「基本XML」と、基本XMLの拡張版である「高度XML」に分けて、作業が行われている。基本XMLは、XMLインスタンスをディスプレイ上に表現するプレゼンテーション形式を、タグを用いて記述することを規定したものである。

【0020】これに対し、高度XMLは、XMLインスタンスの属性情報を付加できるようにしたものである。属性情報の記述方法（すなわちタグの文法）を定めたものが、DTD (Document Type Definition) と呼ばれる規約である。基本XMLではタグが定義する属性情報は固定で、DTDは不要である。これに対し、高度XMLでは、DTDは任意に定義可能であり、例えば各業界毎にDTDを一意に定めることができる。ここで言うタグの属性情報には、例えばテレビ受信機とコンピュータのように異機種間で接続したときに、各機器が互いにコンテンツの意味を理解して処理するために必要な情報などが含まれる。

【0021】XMLインスタンス自体は、表現形式に関するスタイル情報を包含していないこともある。この場合、XMLインスタンスの表現方法は、DTDとは別の文書ファイルである「スタイルシート」によって記述される。

【0022】スタイルシートは、例えば、XMLインスタンスをディスプレイ・スクリーン上に表示出力する形式に変換する（あるいはプリンタへの印刷出力形式に変換する）ための文書ファイルである。XMLインスタンス用のスタイルシートは、特に「XSL (eXtensible Stylesheet Language)」形式の言語、「CSS (Cascade Stylesheet)」形式の言語、あるいは、その派生規格である「XSLT (XSL Transformation)」形式の言語を用いて記述することかできる。スタイルシートの記述次第で、書体やそのサイズ・色などを変化させて、同じDTDコンテンツを全く別の形式で表現することができる。スタイルシートは、DTDとは別のファイルにして送信される。

【0023】なお、スタイルシートの他に、音声や静止画、動画など各モノメディアからなるマルチメディア・コンテンツに関しては、XMLインスタンス又はスタイルシート内で、XMLやXSLとは別のスクリプトと呼ばれる言語によって動作を規定することもできる。このようなスクリプトは、例えばJavaScript、ECMAScript、Jscriptなどのスクリプト言語で記述されるが、XML及びXSL等の規格では、基本的にはECMAScriptで記述するように規定されている（“ECMAScript”は、欧州電子計算機工業会（ECMA）が規格化したスクリプト言語である）。

【0024】XML文書からなる配信コンテンツは、XML文書の他に、例えばDTD文書とスタイルシートを伴う。但し、XML文書がDTD文書を含めない場合や、XML文書の配信とは異なるタイミングでスタイルシートが送付されることもある。また、配信コンテンツの記述言語は、必ずしもXMLには限定されず、SGML (Standard Generalized M

arkup Language) やHTMLなど、その他のマークアップ言語形式であってもよい。また、例えばXML文書がTV放送に適用され、他のリアルタイム型のデータとともに動作してプレゼンテーションを行う場合には、各種のモノメディア・コンテンツのURI (Uniform Resource Identifier) と呼ばれる放送データ・モジュール内のモノメディア位置情報も、XML文書本体に組み込まれることもある。

【0025】受信システム側では、受信したXMLインスタンスを「XMLパーザ (Parser)」と呼ばれる構文解析プログラムでパーズ処理する。すなわち、XMLパーザは、DTD文書及びXMLインスタンスを構造解析して、文書オブジェクトを出力する。この文書オブジェクトは、元のXML文書中のタグに従って木構造が形成された構造化文書である。受信システムの表示機能は、この文書の構造を解釈することで、画面上のレイアウト等を決定し、ディスプレイへの表示出力が可能となる。

【0026】ところで、デジタル・データを放送やネットワーク通信などの伝送手段で配信する場合、XML文書やスクリプトなどのデータ・コンテンツを、元のテキスト形式からバイナリ（2進表示）形式の記述にデータ圧縮してから送信することも考えられる。

【0027】しかしながら、バイナリ形式のファイル構造は、コンピュータ・システムやソフトによって異なるため、アプリケーションや端末間でのコンテンツ交換の障害になりかねない。また、デジタル衛星データ放送の上り回線として、TCP/IP (Transmission Control Protocol/Internet Protocol) ベースで接続されるインターネットも想定している。このインターネットでは、テキスト形式のコンテンツ配信を採用することで、端末の種類やソフトウェアに依存しない環境が築き上げられてきたという歴史的経緯がある。

【0028】そこで、デジタル衛星データ放送においては、インターネットとの親和性やプラットフォーム間のコンテンツ互換性を勘案し、テキスト形式のままデータ・コンテンツを配信することが検討されている。

【0029】ここで問題となるのが、配信コンテンツとしてのテキスト・データのサイズである。何故ならば、データ・サイズに比例して、伝送路の負荷が増大し、伝送効率が低下するからである。

【0030】例えば、XML文書の高機能化又は高付加価値化が進むにつれ、その表現形式や動作を規定するスクリプトの文書サイズは肥大化する。今後、スクリプトが表示対象そのものであるXML文書本体よりもデータ・サイズが大きくなることも予想される。肥大化したスクリプトの配信は、有限な帯域の負荷に与える影響は大きい。

【0031】また、配信コンテンツを受信する受信装置においても、膨大なデータを受信するには大容量のバッファ・メモリを用意する必要があり、コスト増大を招来する。また、メモリ・ロードに要する時間も長くなるため実行効率が低下してしまう。

【0032】受信システムとして汎用コンピュータ・システムを用いる場合、通常、大容量のメモリを標準で搭載しているので、メモリ負荷の問題は比較的小さい。これに対し、セット・トップ・ボックスのような専用の衛星放送受信機の場合、標準メモリのサイズは一般に小さいので、大容量の配信コンテンツは致命的である。

【0033】

【発明が解決しようとする課題】本発明の目的は、テキスト・データからなるコンテンツ配信の伝送効率を向上することができる、優れたデータ配信技術を提供することにある。

【0034】本発明の更なる目的は、テキスト・データからなる配信コンテンツを受信する受信装置における実行効率を向上するとともに、メモリ負荷を低減することができる、優れたデータ配信技術を提供することにある。

【0035】

【課題を解決するための手段】本発明は、上記課題を参酌してなされたものであり、その第1の側面は、配信コンテンツを生成する方法であって、配信コンテンツを構成するモジュールの1つとして作成された複数の文字又は文字列からなるスクリプトの内部を検索して、冗長性の高い文字又は文字列を抽出するステップと、スクリプト中で抽出された冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換えるステップと、を含むことを特徴とする配信コンテンツ生成方法である。前記の冗長性の高い文字又は文字列は、例えば、関数名や変数名などに相当する。

【0036】本発明の第1の側面に係る配信コンテンツ生成方法は、さらに、置換後の配信コンテンツを配信する前に一時蓄積するステップを含んでもよい。

【0037】また、前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換えるステップでは、冗長性の低い文字列としてのシステム予約語の使用を排除するようにしてもよい。システム予約語まで抽出し且つ他の文字又は文字列に置き換えてしまうと、元のスクリプトが本来持つ意味、すなわちスクリプトが規定する動作内容の変更を伴うことになり、許容し難いからである。

【0038】また、該配信コンテンツ生成方法は、さらに、配信コンテンツの受信側で実行される処理に関与しない文字又は文字列からなる文を抽出するステップと、該抽出された文を削除するステップとを含んでもよい。ここで言う処理に関与しない文字又は文字列からなる文は、例えば、所定のデリミッタで区切られたコメント文

に相当する。

【0039】また、該配信コンテンツ生成方法は、さらに、前記の冗長性の高い文字又は文字列を抽出するステップで抽出された文字又は文字列の各々についてのスクリプト中での出現頻度を積算するステップを含み、前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列に置き換えるステップでは、出現頻度の大きい文字又は文字列ほどより文字数の短い文字又は文字列に置き換えるようにしてもよい。出現頻度の大きいもののほど、より短い文字又は文字列で表現されるので、データ圧縮の効果が高まる。

【0040】また、本発明の第2の側面は、複数のデータ・モジュールで構成されるコンテンツを配信するコンテンツ配信方法又は装置であって、複数の文字又は文字列からなるスクリプト言語形式で記述されたデータ・モジュールの内部を検索して、冗長性の高い文字又は文字列を抽出するステップ又は手段と、スクリプト中で抽出された冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換えるステップ又は手段と、置換後のデータ・モジュールを蓄積するステップ又は手段と、蓄積されたデータ・モジュールを配信するステップ又は手段と、を含むことを特徴とするコンテンツ配信方法又は装置である。ここで言う冗長性の高い文字又は文字列とは、例えば、関数名や変数名に相当する。

【0041】また、前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換えるステップ又は手段は、冗長性の低い文字列としてのシステム予約語の使用を排除するようにしてもよい。システム予約語まで抽出し且つ他の文字又は文字列に置き換えてしまうと、元のスクリプトが本来持つ意味、すなわちスクリプトが規定する動作内容の変更を伴うことになり、許容し難いからである。

【0042】また、本発明の第2の側面に係るコンテンツ配信方法又は装置は、さらに、配信コンテンツの受信側で実行される処理に関与しない文字又は文字列からなる文を抽出するステップ又は手段と、該抽出された文を削除するステップ又は手段を含んでもよい。ここで言う処理に関与しない文字又は文字列からなる文は、所定のデリミッタで区切られたコメント文に相当する。

【0043】また、本発明の第2の側面に係るコンテンツ配信方法又は装置は、さらに、前記の冗長性の高い文字又は文字列を抽出するステップで抽出された文字又は文字列の各々についてのスクリプト中での出現頻度を積算するステップ又は手段を含み、前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列に置き換えるステップ又は手段は、出現頻度の大きい文字又は文字列ほどより文字数の短い文字又は文字列に置き換えるようにしてもよい。出現頻度の大きいもののほど、より短い文字又は文字列で表現されるので、データ圧縮の効果が高まる。

10

20

30

40

50

【0044】また、本発明の第3の側面は、コンピュータ処理に関する意味を持つ複数の文字又は文字列で構成されるソース・コードを変換する方法であって、ソース・コード中を検索して、冗長性の高い文字又は文字列を抽出するステップと、抽出された冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換えるステップと、を含むことを特徴とするコード変換方法である。ここで言う冗長性の高い文字列とは、例えば、関数名や変数名に相当する。

【0045】本発明の第3の側面に係るコード変換方法において、前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列で置き換えるステップでは、冗長性の低い文字列としてのシステム予約語の使用を排除するようにしてもよい。システム予約語まで抽出し且つ他の文字又は文字列に置き換えてしまうと、元のスクリプトが本来持つ意味、すなわちスクリプトが規定する動作内容の変更を伴うことになり、許容し難いからである。

【0046】また、本発明の第3の側面に係るコード変換方法は、さらに、ソース・コード中で該コードの実行に関与しない文字又は文字列からなる文を抽出するステップと、該抽出された文を削除するステップとを含んでもよい。ここで言うコードの実行に関与しない文字又は文字列からなる文は、所定のデリミッタで区切られたコメント文に相当する。

【0047】また、本発明の第3の側面に係るコード変換方法は、さらに、前記の冗長性の高い文字又は文字列を抽出するステップで抽出された文字又は文字列の各々についてのスクリプト中での出現頻度を積算するステップ又は手段を含み、前記の冗長性の高い文字又は文字列を冗長性の低い文字又は文字列に置き換えるステップ又は手段は、出現頻度の大きい文字又は文字列ほどより文字数の短い文字又は文字列に置き換えるようにしてもよい。出現頻度の大きいものほど、より短い文字又は文字列で表現されるので、データ圧縮の効果が高まる。

【0048】

【作用】本発明は、デジタル衛星データ放送において、放送番組とともに配信されるデジタル伝送データ、特に、該伝送データのうちテキスト形式で記述されたデータ・ファイルに対して好適に適用することができる。

【0049】〔従来の技術〕の欄で既に説明したように、デジタル伝送データを構成するモジュールは、静止画、動画、音声などの各種様々なモノメディア・データと、これら各々のモノメディア・データを統合的にすなわちマルチメディアとして取り扱い、放送番組情報の有様を規定する表示・出力制御プログラム（以下では、「マルチメディア符号化アプリケーション」とも呼ぶ）とで構成される。

【0050】このモジュール内のファイルのうち、テキスト形式データで構成されるものの1つは、表示・出力制御プログラムであり、そのコンテンツはタグでデータ

属性が定義されるマークアップ記述言語（XMLやSGML、HTMLなど）で記述されている。

【0051】また、該モジュールに含まれる他のテキスト形式データは、この音声や静止画、動画などの各者メディア・データの動作を規定するスクリプトである。スクリプトは、例えばJavaScript、ECMAScript、JScriptなどのスクリプト言語で記述することができる（XML及びXSL等の規格では、基本的にはECMAScriptで記述するように規定されている）。

【0052】XML形式等で記述された表示・出力制御プログラム自体は、画面表示又は印刷出力する情報そのものを規定したものであり、そのテキスト・データの内容変更はデータ自体の意味（すなわち該プログラムが規定する動作）の変化を伴ってしまうので、許容し難い。

【0053】他方、スクリプトの場合、コード中に含まれる関数名や変数名、コメント文などは冗長性を含み、伝送効率上致命的となる。関数名や変数名などは、そもそも、スクリプト作成者が、コーディングやデバッグ作業における可読性などの便宜のため（あるいは個人的な好みに従って）命名したものであり、スクリプト自体が規定する動作機能とは無関係である。言い換えれば、これら関数名や変数名は、スクリプト自体の動作機能を維持したままで、他の文字列との可換性がある。また、コメント文は、スクリプト実行には無関係であり、省略することすら可能である。

【0054】本発明は、このようなテキスト形式で記述されたスクリプトが包含する冗長性に着眼してなされたものである。要するに、本発明は、スクリプト中に存在する、比較的長い文字列からなる関数名や変数名を、より短い文字列（例えばアルファベット1文字）で置き換えることにより、スクリプトのデータ・サイズを削減するものである。また、さらにスクリプト中のコメント文を削除することにより、データ・サイズを削減するものである。

【0055】スクリプト中の関数名や変数名は、スクリプトの作成者が、各々の関数や変数の人手による可読性を保つために便宜上命名したものであり、人が理解可能な意味を持つ程度の比較的長い文字列で構成される。また、コメント文は、スクリプト中の各部分（例えばルーチン文など）の機能をメモするためにソース・コード中に埋め込まれるが、それ自体はスクリプトの機能実現に寄与しない。

【0056】これら長い文字列からなる関数名や変数名、コメント文は、デバックまでの間は意味を持つが、デバック以降は、人が解釈可能な意味を持つ必要はない。特に、衛星データ放送のように一方向性のコンテンツ配信に適用する場合には、送信した以後の段階でのソース・コードの再利用を考慮する必要がない。言い換え

10

20

30

40

50

れば、スクリプト中の関数名や変数名を他の短い文字列に置き換えたり、コメント文を削除して、人がソース・コードの意味内容を解釈し難くなくても、その弊害はほとんどないと言える。

【0057】本発明を、デジタル衛星データ放送におけるデジタル伝送データに適用することにより、伝送データ・サイズを削減することができ、配信時には伝送効率が向上する。

【0058】また、受信局においては、受信データ・サイズが低下するので、比較的小容量の受信メモリを用意するだけで済み、コスト削減につながる。また、スクリプト中の関数名や変数名の文字列長が最小になることから、スクリプトを実行するインタプリタがメモリに割り当てる関数テーブル、変数テーブルのサイズを小さくすることができる。この結果、関数や変数を呼び出す際のオーバーヘッドが少なくなる。すなわち、スクリプトの実行効率が向上する上に、メモリを小容量化して受信機コストを削減することができる。受信機のユーザ（すなわち視聴者）は、実行効率の向上により、より高速なシステムの反応を期待することかできる。

【0059】本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【0060】

【発明の実施の形態】以下、図面を参照しながら本発明の実施例を詳解する。

【0061】図1には、本発明の実施に供される衛星デジタル放送システム1000の概略構成を示している。同図に示すように、衛星デジタル放送システム1000は、放送サービスを提供する放送局（Broadcast Server）1と、放送データを中継する放送衛星（Satellite）5と、放送衛星5から放送データを受信する受信局（Receiver/Decoder）10とで構成される。

【0062】放送局1は地上に1基以上存在し、放送衛星5は地球のはるか上空に散在する。また、受信局10は、一般家庭などに相当し、実際には地上に無数存在する。放送局1及び放送衛星5を介したデータ配信すなわち放送は、一方向通信である。

【0063】デジタル・データ放送では、10～50Mbpsの転送レートでデータ伝送される。放送局1は、放送番組本体を構成するAVデータと、放送番組に付随する番組情報を含んだデジタル伝送データを多重化して配信する。AVデータは、通常、MPEG（Motion Picture Experts Group）2形式でデータ圧縮して送信される。

【0064】デジタル伝送データには、テキストや静止画、動画、音声など各種モノメディア・データと、これらモノメディア・データの表示・出力を制御するプログラムである「マルチメディア符号化アプリケーション」

とで構成されるモジュールが、データ・カラーセル化されて格納されている。マルチメディア符号化アプリケーションは、XML（eXtensible Markup Language）言語形式で記述されており、文書の型式を定義するDTD文書と、XSLなどの言語形式で記述されたスタイルシートを含む（但し、DTD文書及び／又はスタイルシートを配信コンテンツに含まない場合もある）。また、各モジュールには、音声や静止画、動画などの動作を規定するスクリプトが含まれている。

【0065】各受信局10と放送局1との間は、インターネットのような広域ネットワーク7、又は、専用線（図示しない）などによって双方向接続されていてもよい。この場合、インターネット7を、受信局10から放送局1に向かう上り回線として使用することができる。例えば、インターネット7を利用した一部オンデマンド放送サービスを提供することも可能である。但し、この場合、両者間は、56kbps以上の高速アナログ電話回線、10～64kbps程度の高速ワイヤレス通信、128kbpsのISDN（Integrated Services Digital Network）、あるいは5～30Mbpsクラスのケーブルによって接続されていることが好ましい。

【0066】図2には、放送局1すなわち送信システムの構成を模式的に示している。同送信システム1は、制作部100と、送出部200と、伝送部300とで構成される。以下、各部について説明する。

【0067】制作部100は、放送番組のコンテンツを制作する現場に相当する。制作部100は、データ放送の内容を記述したXML文書と、このXML文書の動作を規定するスクリプトと、放送番組の本体であるAVデータなどの情報を制作し、それぞれをローカルの大容量記憶装置102及び103に蓄積する。

【0068】XML文書とスクリプトは、デジタル伝送データを構成する各モジュールのリソース（後述及び図7、図8を参照のこと）に相当する。これらデジタル伝送データは、編集システム101上でコーディング／デバッグ等の編集処理が行われる。編集システム101の実体は、ワークステーションやパーソナル・コンピュータと呼ばれる汎用コンピュータ・システムでよい。

【0069】XML文書は、任意に定義可能なタグを使用したマークアップ言語（前述）であり、文書型を定義するDTD文書を添付してもよい。この場合のXML文書は、受信局10における出力用のデータ本体である。スクリプトは、XML文書の動作を規定したものであり、JavaScript、ECMAScript、JScriptなどのスクリプト言語で記述されたテキスト形式のファイルである（XML及びXSL等の規格では、基本的にはECMAScriptで記述するように規定されている）。なお、デジタル伝送データは、XM

L文書とスクリプト以外に、音声データや静止画データなどのマルチメディア・コンテンツを含んでいてもよい。

【0070】制作部100において制作され蓄積されたこれら配信コンテンツは、例えば、放送局1内に敷設されたLAN (Local Area Network) 経由で、送出部200に転送される。

【0071】送出部200では、コンテンツ送信系201、ベースバンド制御系102及びAVエンコーダ203の各々によって送出データがパケット化されて、伝送部300に渡す。AVエンコーダ203は、放送番組の本体であるAVデータを、例えばMPEG2などの圧縮方式により符号化圧縮する。

【0072】伝送部300では、コンテンツ系のデータはマルチメディア符号化部301において符号化されてコンテンツ伝送系302に渡される。合成部304は、コンテンツ伝送系302及びAVデータ伝送系303の各々の出力データを合成する。そして、変調部305では、合成信号をRF変調して、RF伝送路を介して受信局10へ向けて送信する。

【0073】RF伝送路では、RF信号は、まず、放送局1に設置された送信アンテナから放送衛星5へ送信され、次いで、放送衛星5を経由して受信局10の受信アンテナで受信される。

【0074】図3には、受信局10に設置されるデジタル衛星データ放送受信システムの一例10-Aのハードウェア構成を模式的に示している。該受信システム10-Aは、例えばSTB (セット・トップ・ボックス) と呼ばれる形態で一般家庭に普及している。受信システム10-A内では、メイン・コントローラとしてのCPU11は、バス50を介して各ハードウェア・コンポーネントと相互接続して、各コンポーネントに対して統括的な制御を実行するようになっている。以下、各部について説明する。

【0075】アンテナ (図示しない) で受信された放送波は、チューナ51に供給される。放送波は、規定のフォーマットに従っており、例えば番組ガイド情報 (EPG: Electric Program Guide) 等を含んでいる。なお、放送波としては、上記した放送波以外に、有線放送波や地上波でもよく、特に限定されない。

【0076】チューナ51は、CPU11からの指示に従い、所定チャンネルの放送波のチューニングすなわち選局を行い、後続の復調器52に受信データを出力する。復調器52では、デジタル変調されている受信データを復調する。なお、送信されてくる放送波がアナログかデジタルかに応じて、チューナ11の構成を適宜変更又は拡張することができる。

【0077】復調されたデジタル・データは、MPEG2圧縮されたAVデータと、デジタル伝送データとが多

重化されて構成される「トランスポート・ストリーム」である。前者のAVデータは、放送番組本体を構成する映像及び音声情報である。また、後者のデジタル伝送データは、この放送番組本体に付随するデータであり、例えばEPG (Electric Program Guide: 電子番組ガイド) を含む。デジタル伝送データについては後に詳解する。なお、トランスポート・ストリームは、OSI (Open Systems Interconnection) 参照モデルで言う「トランスポート層」に分類される。

【0078】TSデコーダ53は、このトランスポート・ストリームを解釈して、MPEG2圧縮されたAVデータとデジタル伝送データとに分離して、前者をAVデコーダ54に送出するとともに、後者をバス50経由でCPU11に送信する。TSデコーダ53は、作業データ保管用のメモリ53Aを自己のローカルに備えていてもよい。

【0079】AVデコーダ54は、MPEG2方式で圧縮されたリアルタイムAVデータをTSデコーダ53から受け取ると、圧縮映像データと圧縮音声データとに分離する。そして、映像データに対してはMPEG2伸長処理してもとの映像信号を再生し、音声データに対してはPCM (Pulse Code Modulation) デコードした後に付加音と合成して再生音声信号とする。AVデコーダ54は、作業データ保管用のメモリ54Aを自己のローカルに備えていてもよい。再生映像信号は、マルチプレクサ55Bを介してディスプレイ61に表示出力され、また、再生音声信号は、マルチプレクサ55Aを介してスピーカ62に音声出力される。

【0080】ユーザ・インターフェース制御部56は、ユーザからの入力操作を処理するモジュールであり、例えば、ユーザが直接マニュアル操作するための操作ボタン/スイッチ (図示しない) や、赤外線 (IR) などを介したリモコン66からの遠隔操作を受容する機能を備えている。また、現在の設定内容を表示するための表示パネルやLEDインジケータ (図示しない) を含んでいてもよい。

【0081】ユーザ・インターフェース制御部56が持つ操作ボタン、又は、リモコン66が持つ操作ボタンの1つは、OSC表示コントローラ57からの表示出力 (すなわちデジタル伝送データに基づく番組情報等の表示出力) の有効化/無効化を操作するボタンに割り当てられている。

【0082】CPU (Central Processing Unit) 11は、受信システム10-A全体の動作を統括するメイン・コントローラである。また、CPU11は、バス50経由で転送されてくるデジタル伝送データの処理を行うことができる。デジタル伝送データは、XML (eXtensible Markup Language) 言語形式で記述されており (後

述)、CPU11は、このXML文書に対するXMLパーシングブラウジング(表示出力又は印刷出力)などの処理ソフトウェア(後述)を、オペレーティング・システム(OS)によって提供されるプラットフォーム上で、実行することができる。

【0083】RAM(Random Access Memory)12は、CPU11の実行プログラム・コードをロードしたり、実行プログラムの作業データを書き込むために使用される、書き込み可能な揮発性メモリである。また、ROM(Read Only Memory)13は、受信システム10-Aの電源投入時に実行する自己診断・初期化プログラムや、ハードウェア操作のマイクロコードを恒久的に格納する読み出し専用メモリである。

【0084】シリアル入出力(SIO)コントローラ14は、受信システム10-A外部の機器とシリアル的なデータ交換を行うための周辺コントローラである。SIOコントローラ14が用意するシリアル・ポートには、アナログ電話回線上の伝送データを変復調するための高速モデム63(例えば、転送レートが56Kbps)が外部接続されている。この高速モデム63によって所定のアクセス・ポイント(図示しない)にPPP(Point-to-Point Protocol)接続することで、受信システム10-Aは広域ネットワークとしてのインターネット7に接続される。

【0085】IEEE1394インターフェース15は、数10Mbps程度のデータ送受信が可能なシリアル高速インターフェースである。IEEE1394ポートには、IEEE1394対応の外部機器をダイジェン接続又はツリー接続することができる。IEEE1394対応機器としては、例えば、ビデオ・カメラ64やスキャナ(図示しない)などが挙げられる。

【0086】ハード・ディスク・ドライブ(HDD)17は、プログラムやデータなどを所定フォーマットのファイル形式で蓄積するための外部記憶装置であり、通常、数GB程度の比較的大容量を持つ。HDD17は、ハード・ディスク・インターフェース18を介してバス50に接続される。

【0087】カード・インターフェース18は、カード・スロット19に装填されたカード型デバイス65とバス50との間でのバス・プロトコルを実現するための装置である。カード型デバイス65の一例は、クレジットカード・サイズでカートリッジ式に構成されたPCカードである。PCカードは、PCMCIA(Personal Computer Memory Card Interface Association)及びJEIDA(Japan Electronic Industry Development Association)が共同で策定した仕様"PC Card Standard"に準拠する。

【0088】PCカードの一例は、EEPROM(Electrically Erasable and Programmable ROM)などの不揮発且つ消去再書き込み可能なメモリ・チップで構成されるメモリ・カードである。受信システム10-Aが比較的小型且つ安価に構成される場合、大容量且つ大容積のHDD17を搭載することが、設計上困難な場合がある。このような場合、システム10-Aに対して着脱自在で可搬性に優れたメモリ・カードを適用することが好ましいと想定される。但し、着脱式のメモリ65は、PCカードのフォームファクタに限定されず、所謂「メモリ・スティック」であってもよい。

【0089】表示コントローラ57は、デジタル放送データに基づく放送番組情報等の表示出力を制御するための専用コントローラである。

【0090】このデジタル衛星データ受信システム10-Aでは、CPU11は、ユーザ・インターフェース制御部56を介したユーザ入力コマンドに従って、チューナ51の選局動作を制御するとともに、番組情報の表示制御などが行われる。すなわち、CPU11は、TSデコーダ53から転送されてきたデジタル放送用データを処理して、表示用のデータに変換して表示コントローラ57に供給する。表示コントローラ57は、この表示データを基に番組情報の画像信号を生成し、マルチプレクサ55Bに供給する。また、CPU11は、デジタル放送用データに含まれる音声データも処理して、これをバス50経由でマルチプレクサ55Aに供給する。マルチプレクサ55A及び55Bは、CPU11から供給された表示用データと音声データの各々を、AVデコーダ54から出力される放送番組本体としての映像データ及び音声データと多重化して、ディスプレイ61とスピーカ62の各々に外部出力する。なお、デジタル放送用データの処理については、後に詳解する。

【0091】図4には、他の例に係るデジタル放送データ受信システム10-Bのハードウェア構成を模式的に示している。この例に係る受信システム10-Bは、例えば、汎用コンピュータ・システム30に対して衛星デジタル放送用のデジタル・チューナ・カード40を装備する、という形態で実装される。

【0092】デジタル・チューナ・カード40は、チューナ51と、復調器52と、TSデコーダ53と、AVデコーダ54と、各RAM53A及び54Aと、マルチプレクサ55A及び55Bとで構成される。図3に示したものと同一参照番号を持つハードウェア・ブロックとは、その構成及び機能は略同一である。このデジタル衛星放送用チューナ・カード40は、バス・インターフェース(PCIインターフェース)58(図示)を介してコンピュータ・システム30内のシステム・バス(PCIバス)31に接続される。

【0093】アンテナ(図示しない)から受信された放

送波は、チューナ51で選局され、復調器52で復調される。TSデコーダ53は、トランスポート・ストリームを解釈し、MPEG2圧縮されたAVデータとデジタル伝送データに分離する。AVデータの方はAVデコーダ54に供給され、上述と同様に処理され、ディスプレイ61やスピーカ62などに外部出力される。また、デジタル伝送データの方は、PCIインターフェース58経由でコンピュータ・システム30側に転送され、該システム内部のCPU11によって処理される(後述)。

【0094】他方、汎用コンピュータ・システム30は、CPU11(後述)を始めとする主要回路コンポーネントを搭載した印刷配線基板(図示しない)を含んである。該基板は「マザーボード」とも呼ばれる。先述のチューナ・カード40は、例えば「アダプタ・カード」の形態で提供され、マザーボード上に配設されたバス・スロット(図示しない)に装着される。

【0095】但し、チューナ・カード40は、アダプタ・カードという形態ではなく、PCMCIA(Personal Computer Memory Card International Association)/JEIDA(Japan Electronic Industry Development Association)の策定仕様に準拠したPCカードの形式で提供され、PCカード・スロット19(後述)に挿入することによっても、コンピュータ・システム30のシステム構成に組み込まれる。

【0096】CPU11は、コンピュータ・システム30全体の動作を統括的に制御するメイン・コントローラである。この例に係るCPU11は、オペレーティング・システム(OS)によって提供されるプラットフォーム上で、XMLパーシングやXSLプロセッシング、ブラウジングなど、XMLコンテンツ処理のための各種ソフトウェア・プログラム(後述)を実行することができる。

【0097】CPU11の外部ピンに直結したプロセッサ・バスは、バス・ブリッジ20を介してシステム・バス31に相互接続されている。

【0098】本実施例のバス・ブリッジ20は、プロセッサ・バスとシステム・バス31間の速度差を吸収するためのデータ・バッファの他、RAM12へのメモリ・アクセスを制御するメモリ・コントローラを含んだ構成となっている。

【0099】RAM(Random Access Memory)12は、CPU11の実行プログラム・コードをロードしたり、実行プログラムの作業データを書き込むために使用される、書き込み可能な揮発性メモリである。通常は、複数のDRAM(ダイナミックRAM)チップで構成される。

【0100】システム・バス31は、アドレス・バス、データ・バス、コントロール・バスなどを含んだ共通

号伝送路であり、例えばPCI(Peripheral Component Interconnect)バスがこれに相当する。システム・バス31上には、PCIインターフェース仕様に合致した各種周辺機器が相互接続されている。周辺機器の一例は、先述のデジタル衛星放送用チューナ・カード40である。これらバス31上の各周辺機器にはそれぞれに固有のI/Oアドレス(又はメモリ・アドレス)が割り振られており、CPU11(より厳密にはCPU11が実行するプログラム)は、I/Oアドレス(又はメモリ・アドレス)を指定することで所望の周辺機器に対するデータやコマンドの転送を実現することができる。

【0101】ROM(Read Only Memory)13は、コンピュータ・システム30の電源投入時に実行する自己診断プログラム(POST)や、ハードウェア操作の基本入出力システム(BIOS)を恒久的に格納する読み出し専用メモリである。ROM13は、例えば電氣的な消去及び再書き込み動作が可能なEEPROM(Electrically Erasable and Programmable ROM)で構成されていてもよい。

【0102】シリアル入出力(SIO)コントローラ14は、コンピュータ・システム30外部の機器とシリアル的なデータ交換を行うための周辺コントローラである。SIOコントローラ14が用意するシリアル・ポートには、アナログ電話回線上の伝送データを変復調するための高速モデム63(例えば、転送レートが56Kbps)が外部接続されている。高速モデム63によって所定のアクセス・ポイント(図示しない)にPPP(Point-to-Point Protocol)接続することで、コンピュータ・システム30(すなわち受信システム10-B)はインターネットに接続される。

【0103】IEEE1394インターフェース15は、数10Mbps程度のデータ送受信が可能なシリアル高速インターフェースである。IEEE1394ポートには、IEEE1394対応の外部機器をダイジェネーション接続又はツリー接続することができる。IEEE1394対応機器には、例えば、ビデオ・カメラ64やスキャナ(図示しない)などが挙げられる。

【0104】ハード・ディスク・ドライブ(HDD)17は、プログラムやデータなどを所定フォーマットのファイル形式で蓄積するための外部記憶装置であり、通常、数GB程度の比較的大容量を持つ。HDD17は、ハード・ディスク・インターフェース16を介してシステム・バス33に接続される。ハード・ディスク・ドライブをコンピュータ・システム30に接続するインターフェース規格は、例えばIDE(Integrated Drive Electronics)やSCSI(Small Computer System Interface)などである。

【0105】キーボード/マウス・コントローラ(KMC)21は、キーボード22やマウス23などからのユーザ入力処理するための専用コントローラである。KMC21は、キーボード22からのスキャン・コード入力やマウス23からの座標指示入力を検出したことに応答して、CPU11に対して割り込み要求を発行する。本実施例では、コンピュータ・システム30に対する通常のコマンド入力の他、チャンネルの選択などの受信システム10-Bに対する入力操作も、キーボード22やマウス23を介して行うことができる。

【0106】キーボード22が持つファンクション・キーの1つ、又は、ディスプレイ61上に配設されたメニュー・ボタンの1つは、ビデオ・コントローラ24からの表示出力(すなわちデジタル伝送データに基づく番組情報等の表示出力)の有効化/無効化を操作するボタンに割り当てられている。

【0107】カード・インターフェース18は、バス50とカード・スロット19に装填されたカード型デバイス65との間でのバス・プロトコルを実現するための装置である。カード型デバイス65の一例は、クレジットカード・サイズでカートリッジ式に構成されたPCカードである。PCカードは、PCMCIA(Personal Computer Memory Card Interface Association)及びJEIDA(Japan Electronic Industry Development Association)が共同で策定した仕様"PC Card Standard"に準拠する。

【0108】PCカードの一例は、EEPROM(Electrically Erasable and Programmable ROM)などの不揮発且つ消去再書き込み可能なメモリ・チップで構成されるメモリ・カードである。デジタル衛星放送受信システム10-Bが比較的小型且つ安価に構成される場合、大容量且つ大容量のHDD17を搭載することが、設計上困難な場合がある。このような場合、受信システム10-Bに対して着脱自在で可搬性に優れたメモリ・カードを適用することが好ましいと想定される。但し、着脱式のメモリ65は、PCカードのフォームファクタに限定されず、所謂「メモリ・スティック」であってもよい。

【0109】ビデオ・コントローラ24は、CPU11からの描画命令に従って画面表示を制御する専用コントローラであり、描画情報を一時格納するためのフレーム・メモリ(VRAM)25を備えている。なお、本発明を好適に具現するためには、ビデオ・コントローラ24は、VGA(Video Graphics Array)以上の描画能力(例えば、SVGA(Super video Graphics Array)やXGA(eXtended Graphics Array))を有することが好ましい。

【0110】このデジタル衛星放送受信システム10-Bでは、CPU11は、キーボード22やマウス23を介したユーザ入力コマンドに従って、チューナ11の選局動作を制御するとともに、番組情報の表示制御などを行う。すなわち、CPU11は、TSデコーダ53から供給されたデジタル放送用データを処理して、表示用のデータに変換してビデオ・コントローラ24に供給する。ビデオ・コントローラ24は、この表示データを基に番組情報の画像信号を生成し、マルチプレクサ55Bに供給する。また、CPU11は、デジタル放送用データに含まれる音声データも処理して、これをバス50経由でマルチプレクサ55Aに供給する。マルチプレクサ55A及び55Bは、CPU11から供給された表示用データと音声データの各々を、AVデコーダ54から出力される放送番組本体としての映像データ及び音声データと多重化して、ディスプレイ61とスピーカ62の各々に外部出力する。なお、デジタル放送用データの処理については、後に詳解する。

【0111】なお、衛星デジタル・データ受信システム10を構成するためには、図3や図4に示した以外にも多くの電気回路等が必要である。但し、これらは当業者には周知であり、また、本発明の要旨を構成するものではないので、本明細書中では省略している。また、図面の錯綜を回避するため、図中の各ハードウェア・ブロック間の接続も一部しか図示していない点を了承されたい。

【0112】例えば、図3及び図4には図示しないが、受信システム10は、FD(Floppy Disc)やCD-ROM、MO(Magneto-Optical disc)などの可搬型の記録メディアを装填してデータ・アクセス可能なFDD(Floppy Disc Drive)、CD-ROMドライブ、MOドライブのような外部記憶装置を備えていてもよい。

【0113】図5には、デジタル放送データ受信システム10において実行される各種ソフトウェア・プログラム間における階層的構成を模式的に図解している。以下、各層のソフトウェアの機能について説明する。

【0114】最下層のハードウェア制御層は、オペレーティング・システム(OS)等の上位ソフトウェアに対してハードウェアの相違を吸収する目的を持ち、各ハードウェアに対する直接的な入出力動作やハードウェア割り込みに対応した処理を実行する。

【0115】ハードウェア制御層は、例えばROM13に恒久的に格納されたBIOS(Basic Input/Output System)、あるいは、HDD17にインストールされた「デバイス・ドライバ」という形態で、デジタル衛星放送受信システム10に提供される。

【0116】オペレーティング・システム(OS)は、衛星放送受信システム10内のハードウェア及びソフト

ウェアを総合的に管理するための基本ソフトウェアのことである。OSは、HDD17上のファイルの記録を管理する「ファイル・マネージャ」や、メモリ空間を管理する「メモリ・マネージャ」、システム・リソースの割り振りを管理する「リソース・マネージャ」、タスク実行を管理する「スケジューラ」、ディスプレイ上のウィンドウ表示を制御する「ウィンドウ・システム」などのサブシステムを含んでいる。

【0117】システム・サービスは、アプリケーションなどの上位プログラムがOSに対して各機能呼び出す（コールする）ための関数の集まりであり、API（Application Programming Interface）やラン・タイム・ライブラリ（又はダイナミック・リンク・ライブラリ）がこれに相当する。システム・サービスの存在により、アプリケーションは各ハードウェアを直接操作する必要がなくなり、ハードウェア操作の統一性が担保される。

【0118】XMLアプリケーションは、データ放送の表示・出力などの有様を制御するためのプログラムであり、XMLという任意の定義付けが可能なタグ付の言語で記述されたXML文書である。各XML文書には、文書型定義のためのDTD文書と、表示形式を規定するXSL文書（又は、CSSや、XSLの派生規格であるXSLT）等のスタイルシートが付随してもよい。

【0119】XMLパーザは、DTD文書を用いてXML文書を解析するソフトウェア・プログラムであり、解析結果としての文書オブジェクトをXSLプロセッサに渡す。この文書オブジェクトは、元のXML文書中のタグに従って木構造が形成された構造化文書である。

【0120】次に、衛星データ受信システム10において、放送データを受信する処理手順について、図6～図8を参照しながら説明する。

【0121】デジタル衛星放送システムでは、図6に示すように、まず、放送局1から放送衛星5に対して放送波が送信され、衛星5から受信局10に向かって転送される。この送信放送波のコンテンツは、衛星放送番組本体を構成する映像と音声のデータ（AVデータ）と、この放送番組本体に付随するデジタル伝送データとで構成される。

【0122】図7には、放送波として伝播される放送コンテンツの構成を模式的に示している。同図に示すように、放送コンテンツは、MPEG（Motion Picture Experts Group）2など所定の圧縮方式で圧縮されたAVデータと、デジタル伝送データとを多重化した「トランスポート・ストリーム」として構成される（トランスポート・ストリームは、OSI（Open Systems Interconnection）参照モデルのトランスポート層に分類される）。

【0123】既に述べたように、TSデコーダ53は、

このトランスポート・ストリームを解釈してAVデータ部分とデジタル伝送データ部分とに分離する。AVデータ部分は、AVデコーダ54で処理され、デジタル伝送データ部分はCPU11において処理される。

【0124】デジタル伝送データ部分は、複数のモジュールで構成される。各モジュールは、EPGや広告情報、その他放送番組本体に付随する各種情報を含んでいる。また、該デジタル伝送データ部分は、データ・カルセル（回転木馬式データ）化されており、各モジュールは番組本体の放送中に繰り返し出現する（この結果、視聴者としての受信システム10は、番組放送期間中の任意のタイミングでモジュールを取得することができ、キャッシュ用のメモリを省略できる）。また、MPEG2圧縮されているAVデータの中には、モジュールの表示出力時期との同期をとるための自動開始フラグを埋め込むことができる。

【0125】図8には、モジュールのデータ構造を模式的に示している。同図に示すように、1つのモジュールは、データ放送の表示や出力の有様を規定する制御プログラム（マルチメディア符号化アプリケーション）や、動作を規定するスクリプト、音声、テキスト・データ、静止画、動画などのモノメディア・データなど、複数のリソース要素で構成される。各モノメディア・データは、データ放送の一部を構成するオブジェクトであり、マルチメディア符号化アプリケーションによって統合的に取り扱われる。

【0126】モジュールを構成する各リソース要素は、それぞれ所定フォーマットを持つ独立したファイルである。音声データは、例えばAIFFやWAV、AACなど音声専用のファイル・フォーマットで記述され、静止画は、JPEGやPNG、GIFなど画像専用のファイル・フォーマットで記述される。先頭のリソース・ロケーション情報は、モジュール内における各リソース要素の位置情報を記述している。

【0127】「マルチメディア符号化アプリケーション」は、EPGや広告情報、その他、放送番組本体に付随する各種データ放送の表示・出力の有様を制御するプログラムであり、XML（eXtensible Markup Language）形式で記述されたXMLアプリケーションである。但し、必ずしもXML言語には限定されず、SGMLやHTMLなど、その他のマークアップ言語形式で記述されてもよい。XML文書は、文書型定義を行うDTD文書を含んでもよい。また、XML文書には、現実の表現形式を指定するスタイル情報が含まれていないので、スタイルシートを添付してもよい。

【0128】スタイルシートは、出力コンテンツとしてのXML文書（さらに、その他の音声・静止画などのマルチメディア・コンテンツを含んでもよい）の実際の表現形式を規定したデータ・ファイルである。スタイルシ

ートは、XML文書をディスプレイ61上に表示出力する形式に変換する（あるいはプリンタ（図示しない）への印刷出力形式に変換する）ための、XSL（eXtensible Stylesheet Language）形式で記述された文書ファイルである。

【0129】また、スタイルシートとは別には、音声、テキスト、静止画、動画などの各種モノメディア・データからなるマルチメディア・コンテンツの動作を規定したスクリプトを配信してもよい（スクリプトは、例えばJavaScript、ECMAScript、Jscriptなどのスクリプト言語で記述される。但し、XML及びXSL等の規格では、基本的にはECMAScriptで記述するように規定されている）。

【0130】受信システム10側では、これら受信した配信コンテンツのうち必要なものは、HDD17（又はメモリ・カード65）のような、システム10のローカルな記憶装置に一旦蓄積される。

【0131】衛星放送のように一方向性の送信環境では、標準的なXML言語で記述されたコンテンツは、使用するスタイルシートを限定しない限り、XML文書とスタイルシートが同梱された形態で送信される。

【0132】本実施例では、放送コンテンツのうちのデジタル伝送データ部分は、出力コンテンツ（XML文書）とスタイルシート（XSL文書又はCSS文書など）が同梱されたものの以外に、XML文書本体のみの伝送、XSL文書のみの伝送というケースも想定している。また、XSL文書は、放送以外による配信・配布の形態、例えばインターネットのようなネットワーク経由でのファイル・ダウンロードや、FDやCD-ROM、MO、メモリ・スティックなどの可搬型記憶媒体に担持されて供給を受けることができる。

【0133】また、本実施例に係る受信システム10は、供給されたXML文書とXSL文書（又はCSS文書など）を分離して蓄積する。すなわち、これら供給された文書ファイルを蓄積するHDD17（又は、メモリ・カード65）には、XML文書（但し、DTD文書を含んでもよい）のみを保存・管理する文書本体保存部80と、スタイルシートとしてのXSL文書のみを保存・管理するスタイルシート保存部90が設けられている。

【0134】次に、デジタル衛星放送受信システム10において、放送局1がデジタル伝送データ部分を送信する処理手順について説明する。

【0135】既に上述したように、放送局1が配信するコンテンツは、放送番組本体としてのMPEG2圧縮されたAVデータの他に、デジタル伝送データを含む。図7及び図8を参照しながら説明したように、デジタル伝送データを構成する各モジュールには、テキスト形式データで構成される2つのファイルが含まれている。このうちの1つは出力用データ・ファイルであり、そのコンテンツはタグでデータ属性が定義されるマークアップ記

述言語（XMLやHTMLなど）で記述されている。

【0136】また、モジュールに含まれる他のテキスト形式データは、音声や静止画、動画などの各者メディア・データの動作を規定するスクリプトである。このスクリプトは、例えば、JavaScript、ECMAScript、JScriptなどのスクリプト言語で記述される（XML及びXSL等の規格では、基本的にはECMAScriptで記述するように規定されている）。スクリプトは、ソフトウェアに実行させる処理手順をテキスト形式で記述したものであり、通常、アプリケーションやOS上で、エンド・ユーザが制御できる操作手順を組み合わせた一連の処理を自動化するために利用される。

【0137】XML形式等で記述された出力用データ自体は、画面表示又は印刷出力する情報そのものを規定したものであり、そのテキスト・データの内容変更は許容し難い。

【0138】これに対し、スクリプトのソース・コード中に含まれる関数名や変数名、コメント文などは、冗長性を含み、伝送効率上は致命的となる。関数名や変数名などは、そもそも、スクリプト作成者が、人手による可読性などの便宜上（又は個人的な好みに従って）命名したものであり、スクリプト自体が規定する動作機能とは無関係である。言い換えれば、これら関数名や変数名は、スクリプト自体の動作機能を維持したままで、他の文字列との可換性がある。また、コメント文は、スクリプトの実行とは無関係であり、省略することさえ可能である。

【0139】本発明は、このようなテキスト形式で記述されたスクリプトが包含する冗長性に着眼してなされたものである。要するに、本発明は、スクリプト中に存在する、比較的長い文字列からなる関数名や変数名を、より短い文字列（例えばアルファベット1文字）で置き換えることにより、スクリプトのデータ・サイズを削減するものである。また、さらにスクリプト中のコメント文を削除することにより、データ・サイズを削減するものである。

【0140】図9には、テキスト形式のスクリプト・ファイルを変換する処理ルーチンをフローチャートの形式で示している。このスクリプト・ファイル変換処理ルーチンは、例えば、放送局1内の番組制作者（より具体的には、制作部100の編集システム101（図2を参照のこと）上）において、デジタル伝送データ作成完了後に行われる。以下、このフローチャートの各ステップについて説明する。

【0141】まず、ステップS10では、冗長性の高い文字列と置き換えるための冗長性の低い文字列（以下では、「適用文字」と呼ぶ）と、変数1を初期値に設定する。本実施例では、適用文字として、アルファベット1文字を昇順で使用する。また、適用文字がAから始まっ

てZまで到達すると、AA, AB, AC, ..., ZZ, AAA, ...という具合に、該昇り順に従い文字数を1ずつ増加させることとする。

【0142】また、ECMAScriptには、予約語が小文字から始まるという規則がある。したがって、処理対象となるスクリプトがECMAScript形式で記述されている場合には、適用文字が大文字で始まるように規定することにより、ECMAScriptの予約語との一致を容易に排除することができる。したがって、この場合には、適用文字がスクリプトの予約語と一致するか否かの判定(図12のステップS23:後述)は、必要ない。

【0143】次いで、ステップS12では、処理対象となるスクリプトをスキャンして、冗長性の高い文字列を抽出して、リストアップする。

【0144】ここで言う冗長性の高い文字列とは、例えば関数名や変数名である。これらは、コーディングやデバッグ作業の際の人手による可読性を高めるために、意味を持つ単語を使用した結果として冗長性が高い。例えば、以下に示す規則に従って、関数名や変数名をスクリプト中から抽出することができる。すなわち、

【0145】(1) 予約語"function"の直後に続くワードは関数名である。

(2) 関数名の直後に続く括弧の中身は変数名とカンマの列である。

(3) 予約語"var"の次のワードは変数名である。

(4) 代入された左辺は変数名である。

【0146】次いで、リストの1番目の要素(すなわち元の関数名や変数名)を取り出して(ステップS14)、この要素に対して適用文字を適用して、新しい関数名や変数名とし、要素と適用文字との対応関係を対応表に保存する(ステップS16)。作成された対応表は、後述する[表1]に示す通りである。

【0147】ステップS18では、適用文字がZn、すなわちZだけからなる文字列に到達したか否かを判別する(但し、文字列の個数nをnの冪乗で表記したものである)。

【0148】該判断ブロックの結果が否定的であれば、ステップS20に進んで、アルファベットの昇り順に従い、適用文字を次の文字列に更新する。また、該判断結果が肯定的であれば、ステップS22に進んで、アルファベットの昇り順に従い、適用文字を1文字増やす(例えば適用文字がZであればAAに更新し、ZZであればAAAに更新する)。

【0149】次いで、ステップS24ではiを1だけ増分してリストの次の要素に進み、ステップS26ではリスト中に未処理の要素が残っているか否かを判別する。未処理エントリが残っていれば、ステップS14に復帰して、次の要素に対して上述と同様の処理を繰り返し実行する。また、リスト中の全て要素を対応表に登録し終

えたならば、後続のスクリプト置換処理ルーチン(S30)に進む。該ルーチンでは、元のスクリプト中の関数名や変数名などの冗長文字列を冗長性の低い適用文字に置換して、スクリプトのサイズを圧縮する。そして、この処理ルーチン全体を終了する。

【0150】ステップS30のスクリプト置換処理は、別途定義済みの処理ルーチンであり、図11にはその処理手順をフローチャートの形式で示している。以下、このフローチャートの各ステップについて説明する。

【0151】まず、ステップS31では、対応表([表1]を参照のこと)からエントリを1つ取り出し、次いで、ステップS32では、元のスクリプトから1センテンスを取り出す。但し、ここで言うセンテンスは、ブランクから次のブランク又は次の改行位置までの区間の文字列を指し、ブランクは含まないものとする。

【0152】ステップS33では、センテンス中に、エントリで規定された変数名や関数名が存在するか否かを判別する。判別結果が否定的であれば、次ステップS34をスキップし、逆に肯定的であれば、ステップS34において変数名や関数名を対応する適用文字に置換する。

【0153】次いで、ステップS35では、センテンスがコメント文か否かを判別する。ECMAScript形式のスクリプトであれば、コメント文はデリミッタ"/"/以降で次の改行位置までの文字列である。該判別結果が肯定的であれば、ステップS36に進み、このコメント文をスクリプトから削除する。

【0154】次いで、ステップS37では次のセンテンスの取得を試み、ステップS38では最後のセンテンスであったか否かを判別する。センテンスが未だ残っていれば、ステップS32に復帰して、次のセンテンスに対して上述と同様の処理を繰り返し実行する。

【0155】また、最後のセンテンスに到達していた場合には、対応表中の最後のエントリであったか否かを判別する(ステップS39)。未だエントリが残っていれば、ステップS31に復帰して、次のエントリが規定する対応関係について、上記と同様の処理を繰り返し実行する。また、全ての対応関係について処理し終えていれば、この処理ルーチン全体を終了する。

【0156】このスクリプト変換処理は、制作部100においてスクリプト作成の際に実行して、変換処理後のスクリプトをリソースとして大容量記憶装置102に格納することが好ましい。スクリプト作成直後ではなく、送出直前にスクリプトを変換処理することも考えられるが、この場合、リソースのモジュール・サイズが変更するため、モジュールの再構築が必要となる可能性がある。ゆえに、スクリプト作成時に変換処理を実行しておいた方がよい。(例えば、変換処理前に、モジュール長の制約から1つのモジュールを2つ以上のモジュールに分割して格納していたような場合、変換処理により単一

のモジュール長に収まることもある。)

【0157】図10には、図9に示した変換処理を実行する前後のスキプットのソース・コードを対比している。但し、このスキプットは、ECMAScript (ECMAScriptは、欧州電子計算機工業会が規格化したスキプット言語である)で記述されたテキスト・データであり、また、完結したプログラムではなくその一部の抜粋である点を了解されたい。

【0158】元のソース・コード中で、"function"なる予約語の直後に続く文字列(例えば、"buildArray"など)は関数名であり、より短い文字列に置き換える対象となる。

【0159】また、代入された左辺の文字列(例えば、"temporary_argument"など)は変数名に該当し、同様に短い文字列に置き換える対象となる。スキプット内で定義された変数を挙げたが、本*

*発明の要旨はこれに限定されない。例えば、XMLやHTMLなどの文書中の表示用符号化データの属性を操作する操作子としての変数にも、同様に、短い文字列への置換処理を適用することができる。

【0160】また、デリミッタ"/"で始まる行末までの文字列は、コメント文である。本実施例に係る置換処理では、ソース・コード中で、"/"から始まる1文を検索して、その行末までの文字を全て無視すなわち削除する。

【0161】図10に示す元のソース・コードに対して図9に示す変換処理を実行した結果、以下の【表1】に示す対応表が生成される。そして、抽出された変数名や関数名などの冗長性の高い文字列は、冗長性の低い適用文字に変換される。

【0162】

【表1】

変換後文字列	変換前文字列(原関数名、原変数名)
A	buildArray
B	temporary_argument
C	i
D	urls
E	go
F	which
G	number
H	resident
I	Index
J	url

【0163】図10を参照して判るように、元のソース・コード中の変数名や関数名は、アルファベット1文字に置き換えられ、コメント文は削除される。また、同図で視認される通り、コード量は大幅に削減され、伝送効率は大いに向上する。

【0164】変換処理後も、スキプットが規定する処理そのものは変換前と全く同じである。したがって、変換処理されたスキプットを受信する衛星放送受信システム10では、スキプットの実行系はこのサイズ削減処理がなされたか否かを意識せず同じように取り扱うことができる。

【0165】スキプット中の関数名や変数名の文字列長が最小になることは、受信システム10における実行効率の観点からも好ましい。何故ならば、スキプットを実行するインタプリタがメモリに割り当てる関数テーブル、変数テーブルのサイズを小さくすることができる。この結果、関数や変数を呼び出す際のオーバーヘッドが少なくなるからである。

【0166】図9に示すスキプット変換処理ルーチンによれば、可読性を持つゆえに冗長であった関数名や変数名を、冗長性を排除した短い文字列すなわち適用文字に変換することで、スキプットのサイズを好適に圧縮することができる。

【0167】置換対象となる関数名や変数名の個数が26以下、すなわち、アルファベット文字の総数以下であれば、いかなる順序で適用文字への置換処理を行っても、変換後のサイズに相違はない。

【0168】これに対し、置換対象が26個を越える場合には、1文字の適用文字の他に、2文字以上の適用文字を使用することになる(図9のステップS22を参照のこと)。より長い文字列を短い文字列に置換した方が、データ圧縮の効果は高い。

【0169】図12には、図9に示したものと別のスキプット変換処理ルーチンの例をフローチャートの形式で示しているが、該フローチャートではこのデータ圧縮効率の点を勘案している。すなわち、ステップS12とステップS14の間に、リスト中の要素を再ソートするステップS13を挿入した。すなわち、このステップS13では、スキプット中での出現頻度が大きい順にリスト中の各要素を再ソートする。この結果、以後の置換処理では、出現頻度の大きい文字又は文字列ほど、より文字数の短い文字又は文字列に置き換えられることになり、データ圧縮の効果が高まる。

【0170】また、図12に示すルーチンでは、さらに、ステップS20とS24の間に、更新された適用文字がシステム予約語か否かを判断し(ステップS2

3)、予約語であればステップS18に復帰して文字列変換の対象から除外するようにしている。予約語に変換してしまうと、元のスクリプト本来の意味、すなわちスクリプトが規定する動作が変更してしまうため、許容し難いからである。

【0171】スクリプト中の関数名や変数名は、スクリプトの作成者が、各々の関数名や変数名の人手による可読性を保つために便宜上命名したものであり、人が理解可能な意味を持つ比較的最長い文字列で構成される。また、コメント文は、スクリプト中の各部分（例えばルーチン文など）の機能をメモするためにソース・コード中に埋め込まれるが、それ自体は機能実現に寄与しない。

【0172】これら長い文字列からなる関数名や変数名、コメント文は、デバックまでの間は意味を持つが、デバック以降は、人が解釈可能な意味を持つ必要はない。特に、衛星放送のように一方向性のコンテンツ配信に適用する場合には、送信した以後の段階でのソース・コードの再利用を考慮する必要がない。言い換えれば、スクリプト中の関数名や変数名を他の短い文字列に置き換えたり、コメント文を削除して、ソース・コードが人手での可読性を失っても、その弊害はほとんどないのである。

【0173】本発明を、デジタル衛星データ放送におけるデジタル伝送データに適用することにより、伝送データ・サイズを削減することができ、配信時には伝送効率が向上する。

【0174】また、受信局においては、受信データ・サイズが低下するので、比較的小容量の受信メモリを用意するだけで済み、コスト削減につながる。また、関数名や変数名の文字列長が最小になることから、スクリプトを実行するインタプリタがメモリに割り当てる関数テーブル、変数テーブルのサイズを小さくすることができる。この結果、関数や変数を呼び出す際のオーバーヘッドが少なくなる。すなわち、スクリプトの実行効率が向上する上に、メモリを小容量化して受信機コストを削減することができる。

【0175】[追補]以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

【0176】

【発明の効果】以上詳記したように、本発明によれば、テキスト・データからなる配信コンテンツの伝送効率を向上することができる、優れたデータ配信技術を提供することができる。

【0177】また、本発明によれば、テキスト・データ

からなる配信コンテンツを受信する受信装置における実行効率を向上するとともに、メモリ負荷を低減することができる、優れたデータ配信技術を提供することができる。

【0178】本発明をデジタル衛星放送などのデジタル・データの配信に適用した場合、テキスト形式で記述されたスタイルシートを、その処理内容を変えずにデータ量を削減することができる。この結果、コンテンツ配信元である放送局においては、スタイルシートの作成作業は従来のままで、配信コンテンツの伝送効率を向上させることができる。

【0179】また、受信局側においては、受信データ・サイズが低下するので、比較的小容量の受信メモリを用意するだけで済み、コスト削減につながる。

【0180】さらに、スクリプト中の関数名や変数名の文字列長が最小になることから、スクリプトを実行するインタプリタがメモリに割り当てる関数テーブル、変数テーブルのサイズを小さくすることができる。この結果、関数や変数を呼び出す際のオーバーヘッドが少なくなり、スクリプトの実行効率が向上する。したがって、受信機のユーザ（すなわち視聴者）は、実行効率の向上により、より高速なシステムの反応を期待することかできる。

【0181】また、スクリプトを配信する放送局側では、変換前後でのスクリプトの意味・内容は変化させていないので、受信側では既存の解釈ソフトウェアを用いてスクリプトを駆動することができる。言い換えれば余分なソフトウェア開発を必要としない。

【図面の簡単な説明】

【図1】本発明の実施に供される衛星デジタル放送システム1000の概略構成を示した図である。

【図2】放送局1すなわち送信システムの構成を模式的に示した図である。

【図3】受信局10（すなわち一般家庭）における受信システムの一例10-Aのハードウェア構成を模式的に示した図であり、より具体的には、STB（セット・トップ・ボックス）と呼ばれる形態で実装された受信システム10-Aの構成を示した図である。

【図4】他の例に係る受信システム10-Bのハードウェア構成を模式的に示した図であり、より具体的には、汎用コンピュータ・システムに衛星デジタル放送用のチューナ・カードを装備するという形態で実装された受信システム10-Bのハードウェア構成を示した図である。

【図5】デジタル放送データ受信システム10において実行される各種のソフトウェア・プログラムにおける階層的構成を模式的に図解している。

【図6】放送コンテンツが放送局1から放送衛星5を介して受信局10に転送される様子を模式的に示した図である。

【図7】放送波として伝播される放送コンテンツの構成を模式的に示した図である。

【図8】デジタル伝送データを構成するモジュールのデータ構造を模式的に示した図である。

【図9】スクリプトを変換する処理ルーチンを示したフローチャートである。

【図10】図9に示した変換処理を実行する前後のスクリプトのソース・コードを対比した図である。

【図11】別途定義されたスクリプト置換処理ルーチン(ステップS30)を示したフローチャートである。

【図12】スクリプトを変換する処理ルーチンの他の例を示したフローチャートである。

【図13】デジタル伝送データとして配信コンテンツのデータ構造を模式的に示した図である。

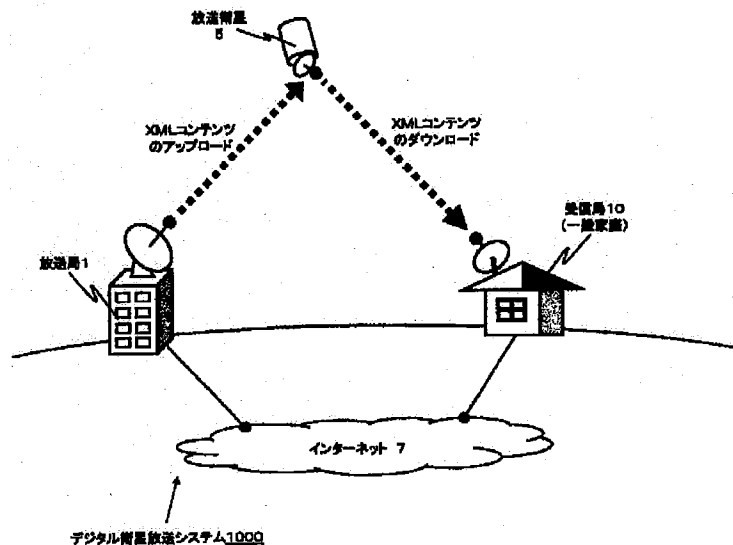
【符号の説明】

1…放送局(デジタル放送データ送信システム)
5…放送衛星, 7…広域ネットワーク(インターネット)
10…受信局(一般家庭: デジタル放送データ受信システム)
11…CPU, 12…RAM(Random Access Memory)
13…ROM(Read Only Memory),
14…SIOコントローラ
15…IEEE1394コントローラ
16…ハード・ディスク・インターフェース
17…ハード・ディスク・ドライブ(HDD)
18…カード・インターフェース, 19…カード・スロット

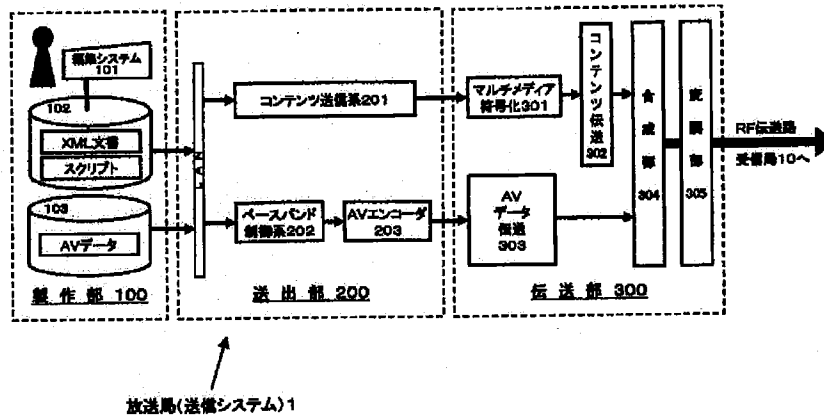
* ユニット

20…バス・ブリッジ
21…キーボード/マウス・コントローラ(KMC)
22…キーボード, 23…マウス
24…ビデオ・コントローラ, 25…VRAM
30…汎用コンピュータ・システム
31…システム・バス(PCIバス)
40…デジタル・チューナ・カード
50…バス, 51…チューナ, 52…復調器, 53…TSデコーダ
54…AVデコーダ
53A, 54A…RAM(Random Access Memory)
55A, B…マルチプレクサ(MUX)
56…ユーザ・インターフェース制御部
57…表示コントローラ, 58…バス(PCI)・インターフェース
61…ディスプレイ, 62…スピーカ
63…高速モデム, 64…ビデオ・カメラ, 65…メモリ・カード
66…リモコン
100…制作部, 101…編集システム
102, 103…大容量記憶装置
200…送出部, 201…コンテンツ送信系
202…ベースバンド制御系, 203…AVエンコーダ
300…伝送部, 301…マルチメディア符号化部
302…コンテンツ伝送部, 303…AVデータ伝送部
304…合成部, 305…変調部
1000…デジタル衛星データ放送システム

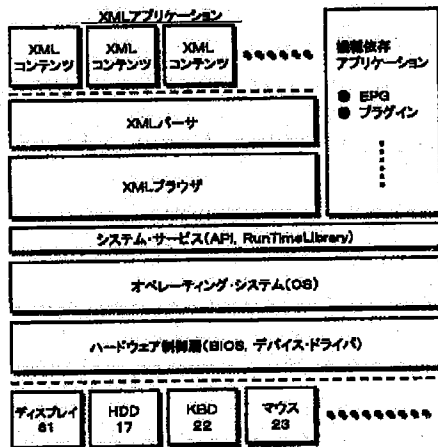
【図1】



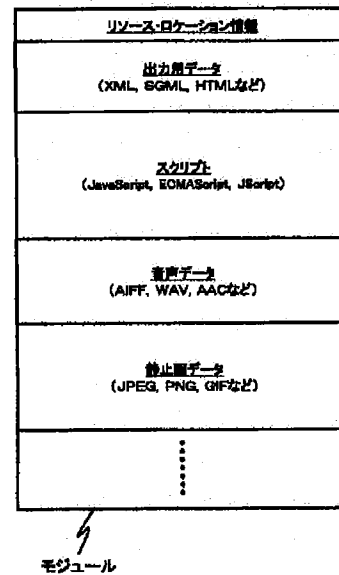
【図2】



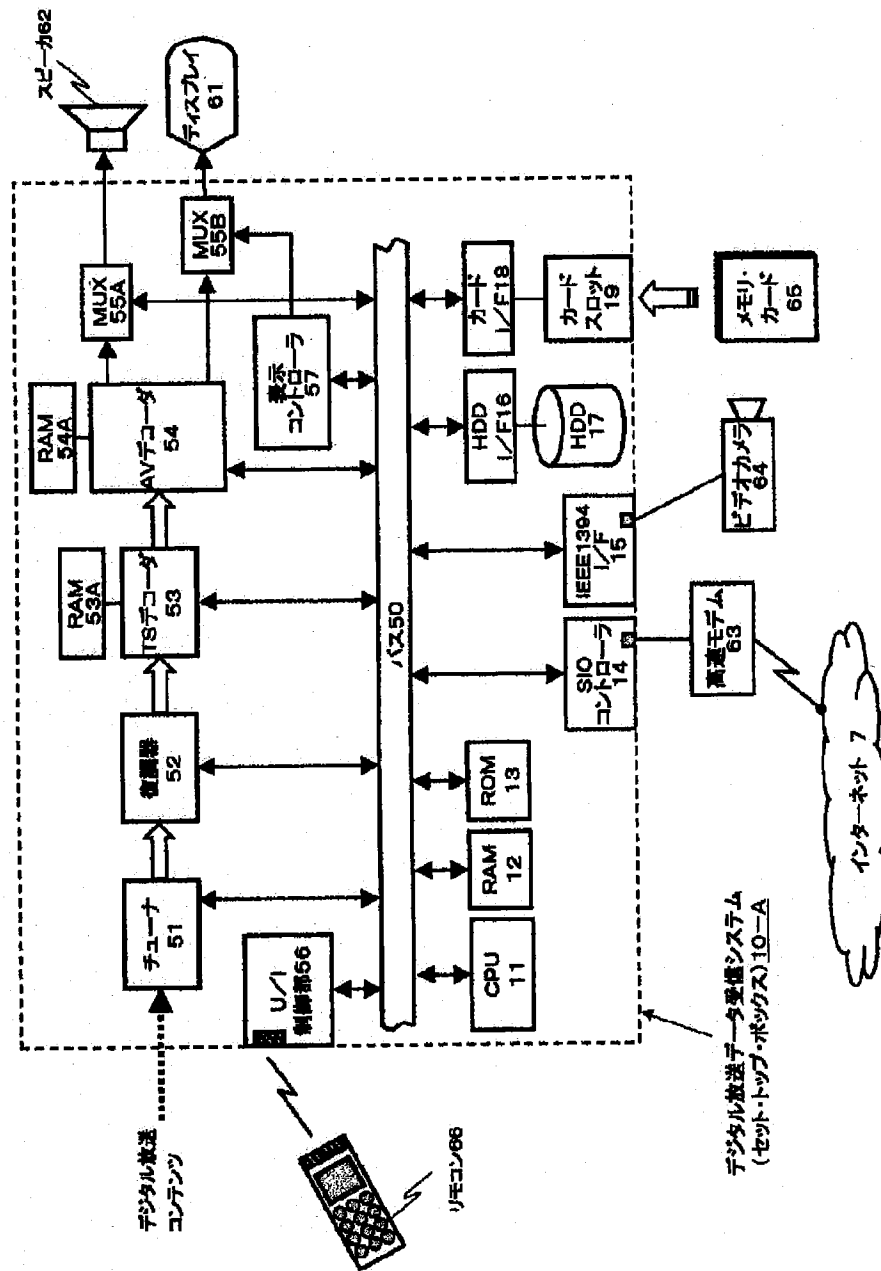
【図5】



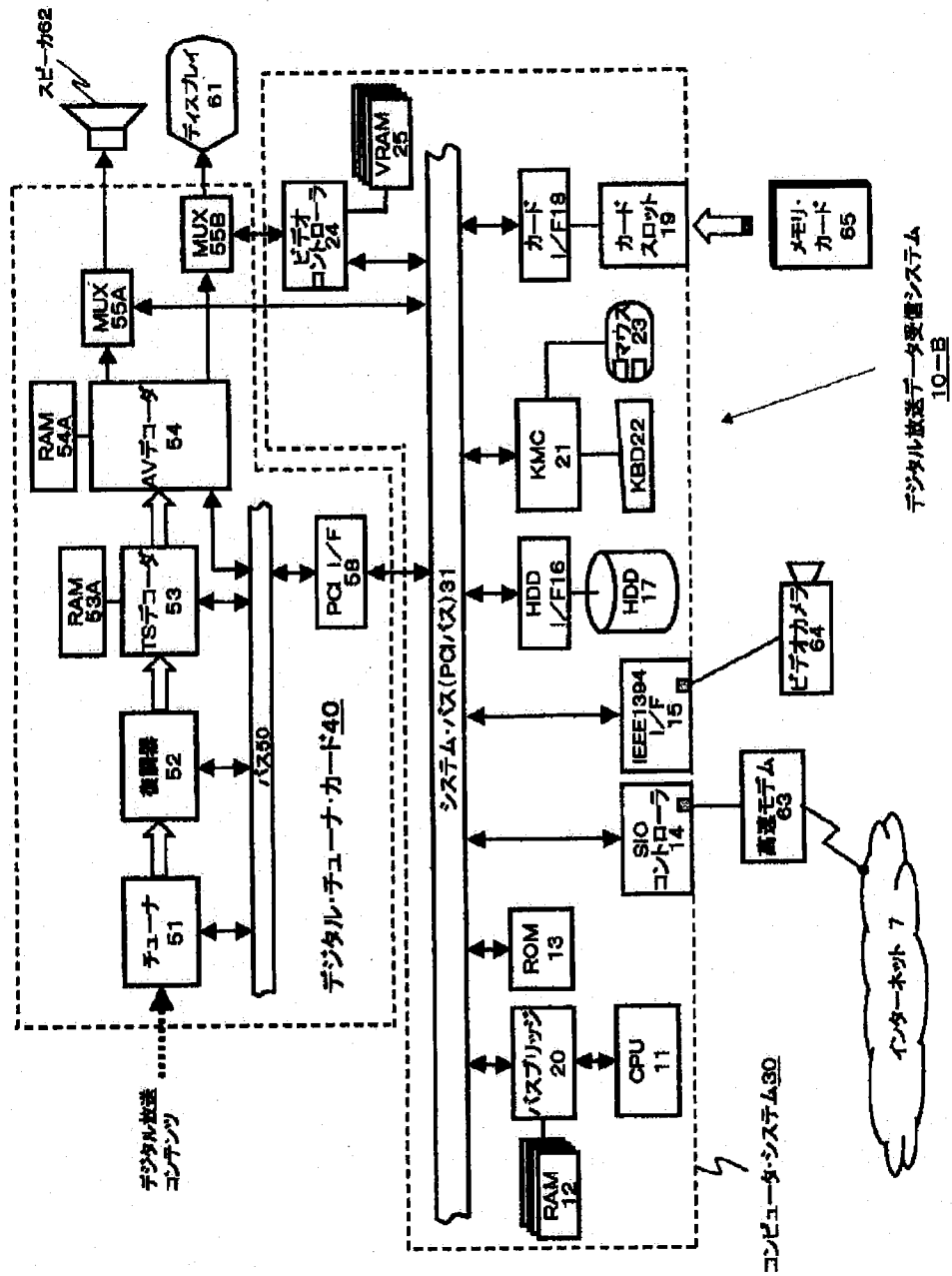
【図8】



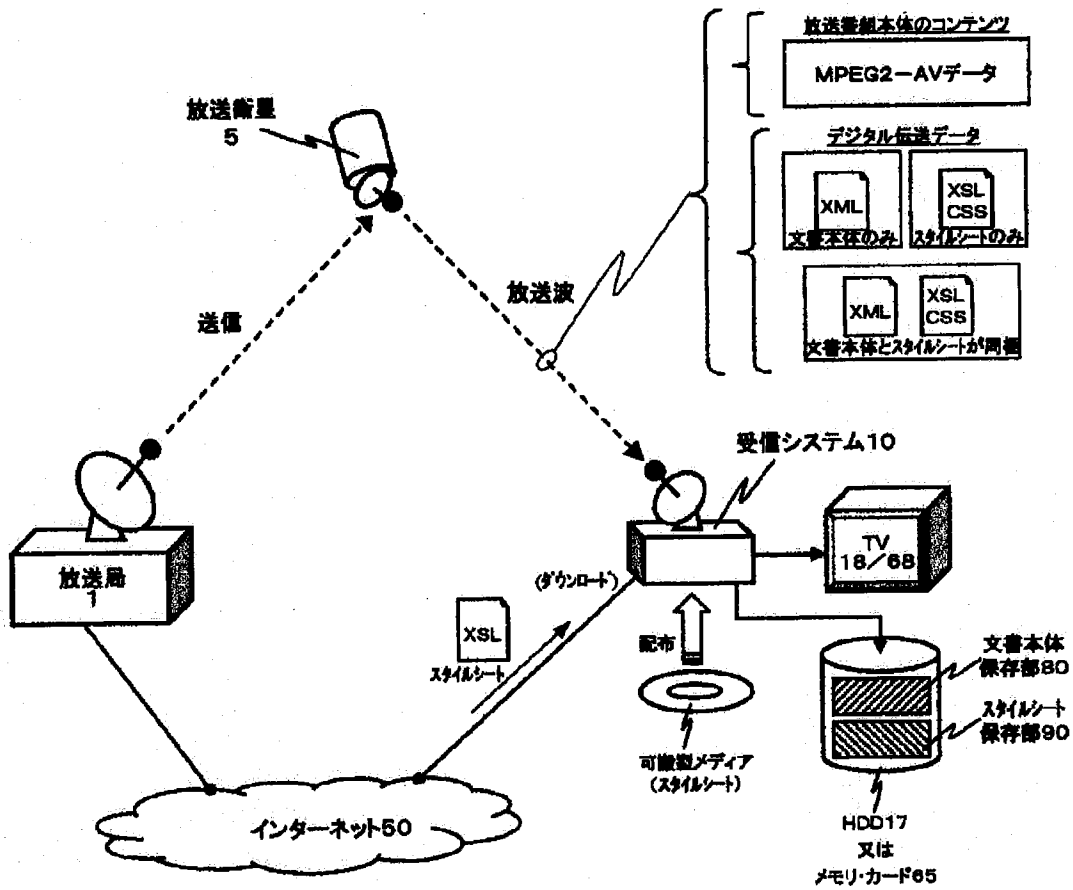
【図 3】



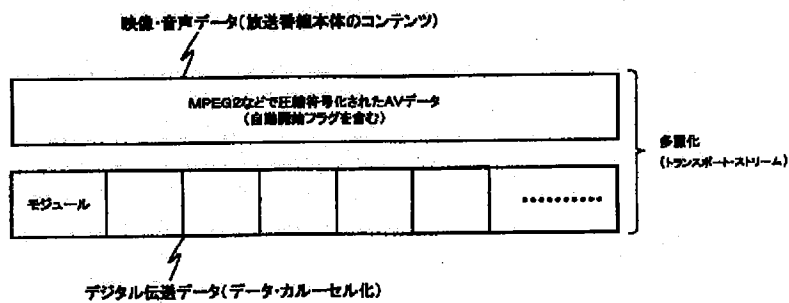
【図4】



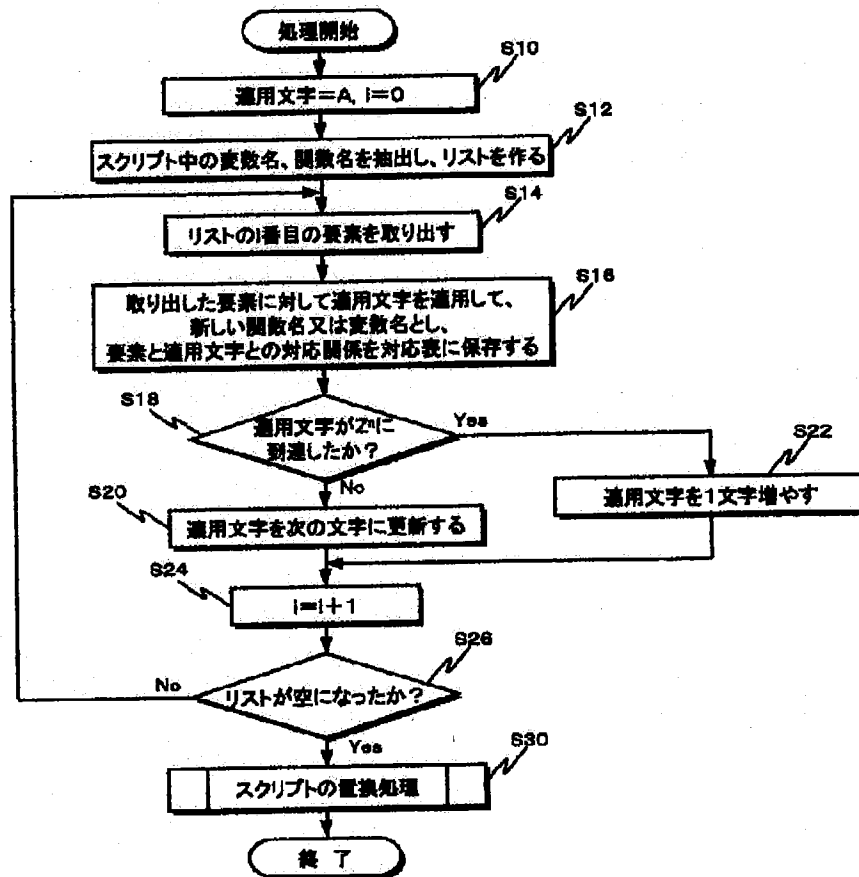
【図6】



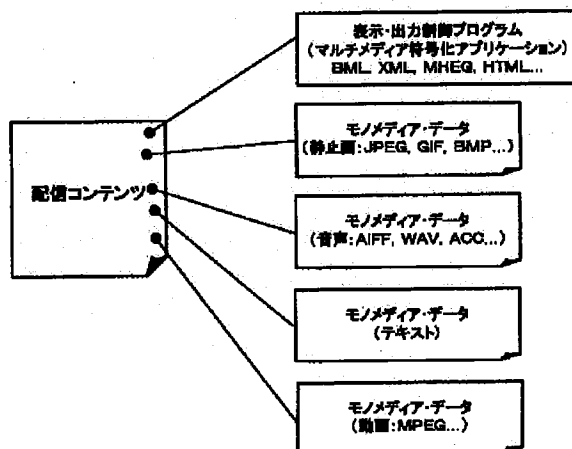
【図7】



【図9】



【図13】



【図10】

```

function buildArray0      //配列の生成
{
    var temporary_argument = buildArray.arguments;
    // 引数を変数に入れる。
    for (i = 0; i < temporary_argument.length; i++){
        //変数の長さ分繰り返して
        this[i] = temporary_argument[i];
        //
    }
    this.length = temporary_argument.length;
}

var urls = new buildArray("http://www.foo.bar/xxxx", "http://www.xx.yy.zz/foo
sample");

function go(which, number, resident)
{
    Index = which.selectedIndex;
    if (Index != 0)
    {
        var url = getUrl(urls, number, Index); //組みこみ関数
        if (resident) {
            gotoURL(url); // 組みこみ関数
        } else {
            location.href = url;
        }
    }
}

```

↑

```

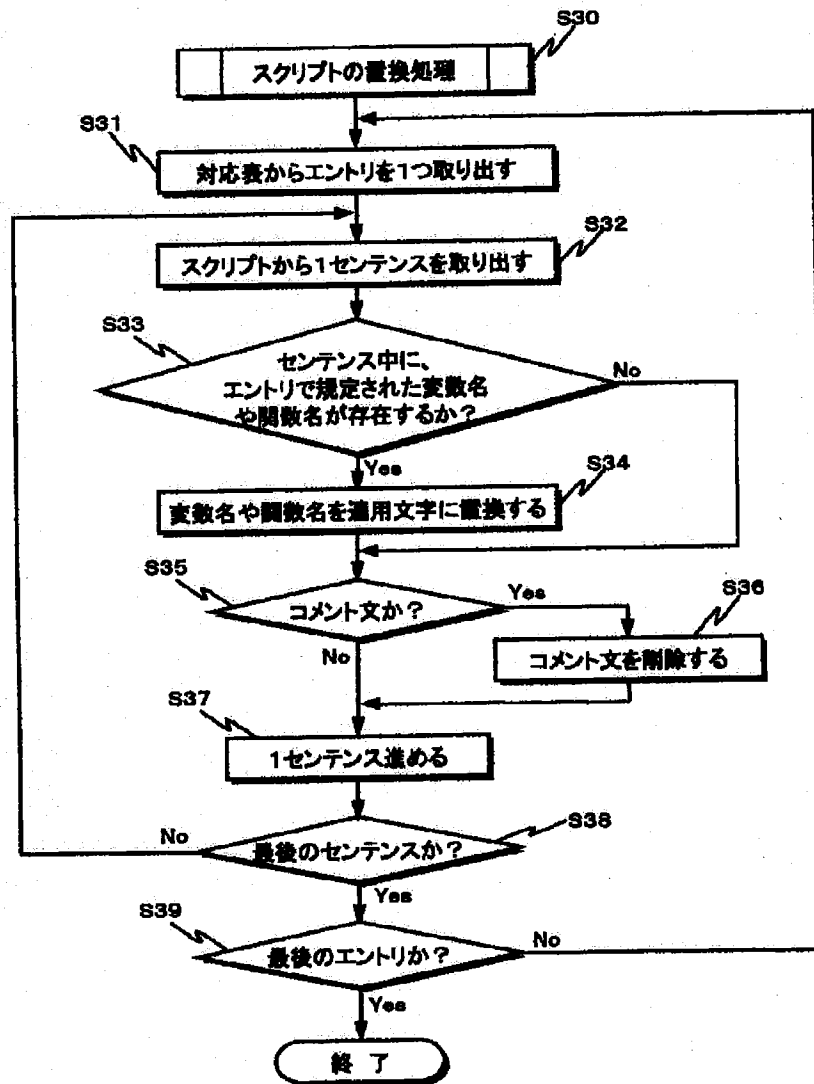
function A0
{
    var B = A.arguments;
    for (C = 0; C < B.length; C++){
        this[C] = B[C];
    }
    this.length = B.length;
}

var D = new A("http://www.foo.bar/xxxx", "http://www.xx.yy.zz/foo
sample");

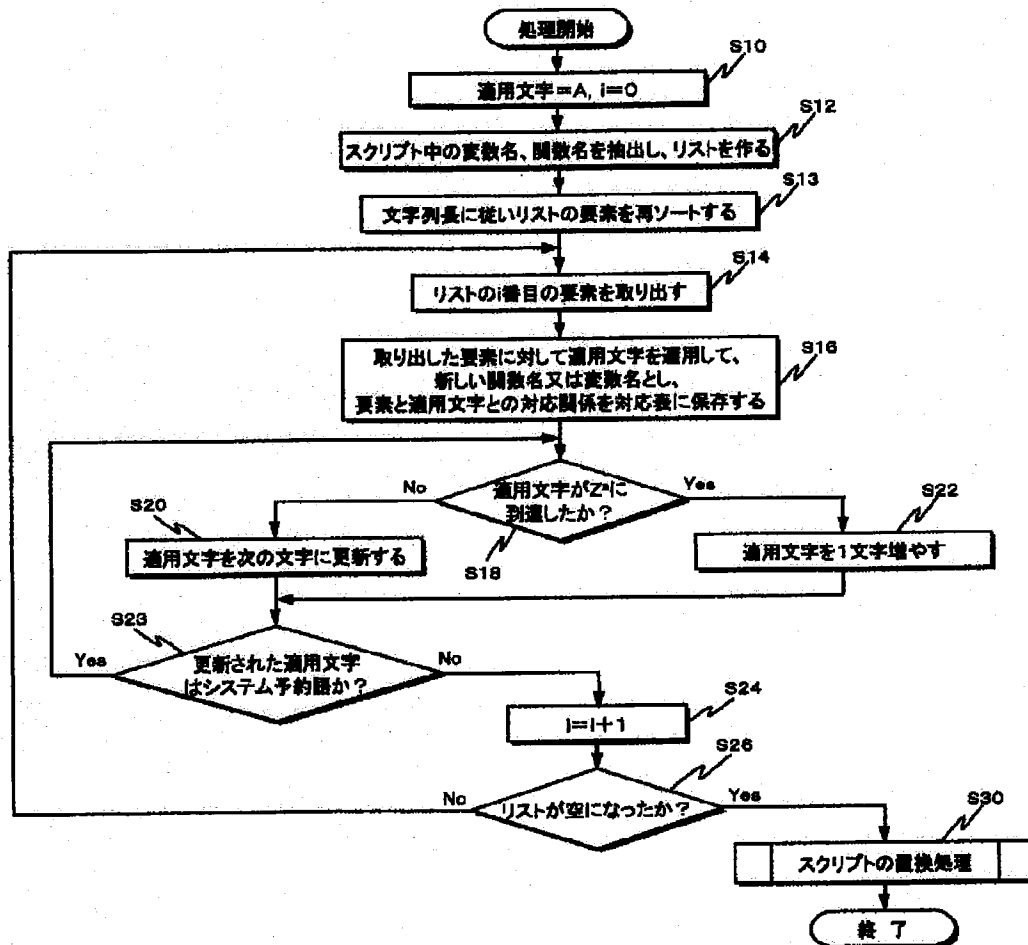
function E(F, G, H)
{
    I = F.selectedIndex;
    if (I != 0)
    {
        var J = getUrl(D, G, I);
        if (H) {
            gotoURL(J);
        } else {
            location.href = J;
        }
    }
}

```

【図11】



【図12】



フロントページの続き

F ターム(参考) 5B082 GA02 HA05
 5B089 JA33 JB02 JB08 KA06 KA08
 KC44 KC51 KC53 KC54 KH23
 KH28
 5C064 BA01 BA07 BB05 BB10 BC10
 BC16 BC20 BC27 BD07 BD08
 BD09 BD13
 5K030 HB21 LD07 LD13
 9A001 BB04 CC08 DD09 DD10 EE04
 JJ19 JJ20 JJ25 JJ67 KK60